DD2434 – Advanced Machine Learning

Lecture 4: Kernels and introduction to Gaussian processes

Pawel Herman

Computational Science and Technology (CST)

KTH Royal Institute of Technology

Short outline for today

1. Recap from Lecture 3
   a) Bayesian linear regression
   b) evidence framework
2. Kernel methods
   a) dual linear regression
   b) kernel functions
3. Gaussian processes, part I

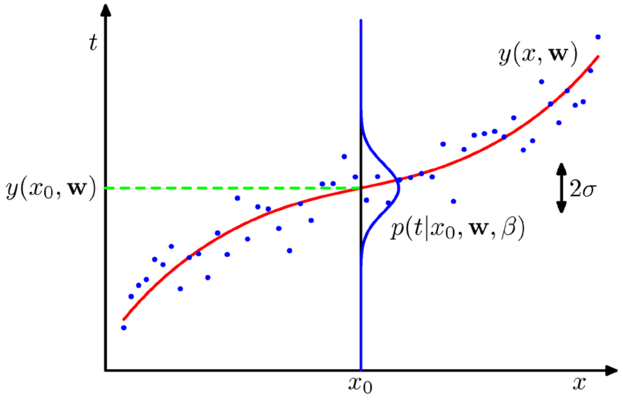
Linear regression – the "*work horse*" of ML

- Linear basis function models

$$\mathbf{x} \rightarrow y: \; y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$$

$$\mathbf{w} = \left(w_0, \ldots, w_{M-1}\right)^{\mathrm{T}} \qquad \boldsymbol{\phi} = \left(\phi_0, \ldots, \phi_{M-1}\right)^{\mathrm{T}}$$

e.g., for polynomial one-dimensional regression basis functions are $\phi_j(x) = x^j$



Linear regression – the "*work horse*" of ML

- Linear basis function models

$$\mathbf{x} \to y: \; y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x})$$

$$\mathbf{w} = \left(w_0, \ldots, w_{M-1}\right)^{\mathrm{T}} \qquad \boldsymbol{\phi} = \left(\phi_0, \ldots, \phi_{M-1}\right)^{\mathrm{T}}$$

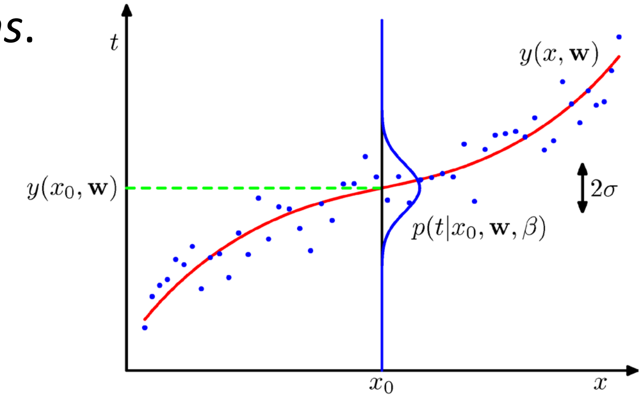
e.g., for polynomial one-dimensional regression basis functions are $\phi_j(x) = x^j$

Ubiquitous **uncertainty:** *observations*, *mapping*, *predictions*.

$$t = y(\mathbf{x},\mathbf{w}) + \varepsilon = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) + \varepsilon$$

$$p(t \mid \mathbf{x},\mathbf{w},\beta) = \mathcal{N}(t \mid y(\mathbf{x},\mathbf{w}), \beta^{-1})$$

"Noise" model (CLT)



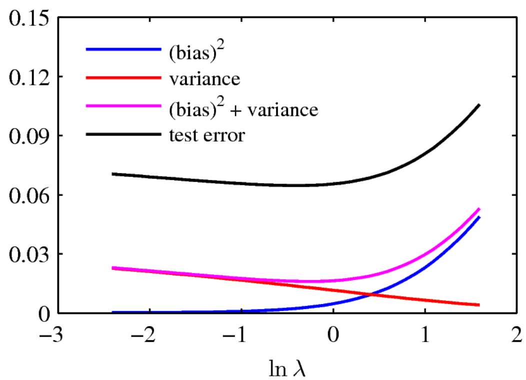
Bayes and linear regression

- General philosophy of a Bayesian approach

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

- In the maximum likelihood formulation however, we run into…



increased model complexity

$$\mathrm{E}[L] = (\text{bias})^2 + \text{variance} + \text{noise}$$

Linear regression – maximum likelihood

- Maximum likelihood estimation

$$y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$$

Likelihood:

$$p(\mathbf{t}_{\mathcal{D}_{trn}} \mid \mathbf{X}_{\mathcal{D}_{trn}}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n), \beta^{-1})$$

t conditionally independent of $\mathbf{x}$

Let's maximise the likelihood or log-likelihood:

$$\Downarrow$$

$$\max_{\mathbf{w}} \arg\left\{\ln p(\mathbf{t}_{\mathcal{D}_{trn}} \mid \mathbf{X}_{\mathcal{D}_{trn}}, \mathbf{w}, \beta)\right\}$$

$$\ln p(\mathbf{t}_{\mathcal{D}_{trn}} \mid \mathbf{X}_{\mathcal{D}_{trn}}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$$

Linear regression – maximum likelihood

- Maximum likelihood $\left(\frac{\partial \ln p(\mathbf{w} \mid \mathbf{X}, \mathbf{t})}{\partial \mathbf{w}} = 0 \right)$

$$0 = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\mathbf{x}_n) - \mathbf{w}^{\mathrm{T}} \left(\sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \right)$$

Least-square solution:
(normal equations)

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}$$

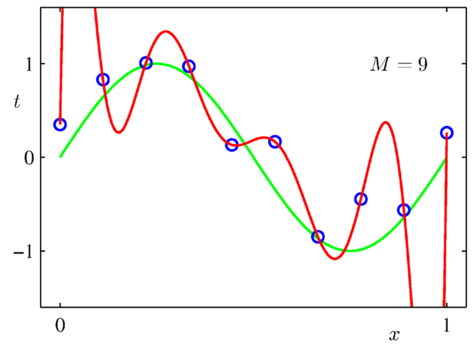
pseudo-inverse of the design matrix $\boldsymbol{\Phi}$

$$\beta_{\mathrm{ML}}^{-1} = \frac{1}{N} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

Linear regression – regularisation

- Problems with maximum likelihood estimate



- We can address it by regularisation (parameter shrinkage)

$$\arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 + \frac{\lambda}{2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \right\}$$

(weight decay)

$$\arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|^q \right\}$$

(lasso for q=1)

Bayes and linear regression

➢ General philosophy of a Bayesian approach

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

- Uncertainty in outputs: additive Gaussian distributed noise

$$t = y(\mathbf{x},\mathbf{w}) + \varepsilon = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + \varepsilon, \quad \varepsilon \in \mathcal{N}(0,\sigma^2)$$ *Central Limit Theorem*

- The *observations*: likelihood $p(\mathbf{t} \mid \mathbf{w}, \mathbf{X})$ (is there any cond. (in)dependency?)
- Express *belief* (prior) about the model before any data are seen
- Posterior (conditional distribution) updated after relevant information has been taken into account

Bayesian linear regression – prediction

- Predictive distribution (with conjugate Gaussian prior)

$$\underbrace{p(t \mid \boldsymbol{x}, \boldsymbol{t}_{\mathcal{D}_{trn}}, \mathbf{X}_{\mathcal{D}_{trn}}, \alpha, \beta)}_{\text{predictive}} = \int \underbrace{p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta)}_{\text{"noise" model}} \; \underbrace{p(\boldsymbol{w} \mid \boldsymbol{t}_{\mathcal{D}_{trn}}, \mathbf{X}_{\mathcal{D}_{trn}}, \alpha, \beta)}_{\text{posterior}} \, \mathrm{d}\boldsymbol{w}$$

$$p(t \mid \boldsymbol{x}, \boldsymbol{t}_{Dtrn}, \mathbf{X}_{\mathcal{D}_{trn}}, \alpha, \beta) = \mathcal{N}(t \mid \boldsymbol{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}), \sigma_N^2(\boldsymbol{x}))$$

$$\boldsymbol{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{t}_{\mathcal{D}_{trn}}$$

$$\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}$$

$$\sigma_N^2(\boldsymbol{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\boldsymbol{x})$$

Bayesian model comparison

- Evidence framework

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

  - The denominator does not change with $\mathbf{w}$

$$p(\mathbf{w} \mid \mathbf{t}_{\mathcal{D}}, \mathbf{X}_{\mathcal{D}}) \propto p(\mathbf{t}_{\mathcal{D}} \mid \mathbf{w}, \mathbf{X}_{\mathcal{D}})\, p(\mathbf{w})$$

  - $p(\mathcal{D})$ shows where the model spreads its probability mass over the data space (evidence of the model)

$$p(\mathbf{w} \mid \mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D} \mid \mathbf{w}, \mathcal{M}_i)\, p(\mathbf{w} \mid \mathcal{M}_i)}{p(\mathcal{D} \mid \mathcal{M}_i)}$$

Bayesian model comparison

- Evidence framework

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})\, p(\mathbf{w})}{p(\mathcal{D})}$$

➢ Th

➢ p

th

over

$$p(\mathcal{D} | \mathcal{M}_i) = \int p(\mathcal{D} | \mathbf{w}, \mathcal{M}_i)\, p(\mathbf{w} \mid \mathcal{M}_i)\, d\mathbf{w}$$

"The evidence can be seen as the probability of generating the data set from a model whose parameters are sampled at random from the prior"

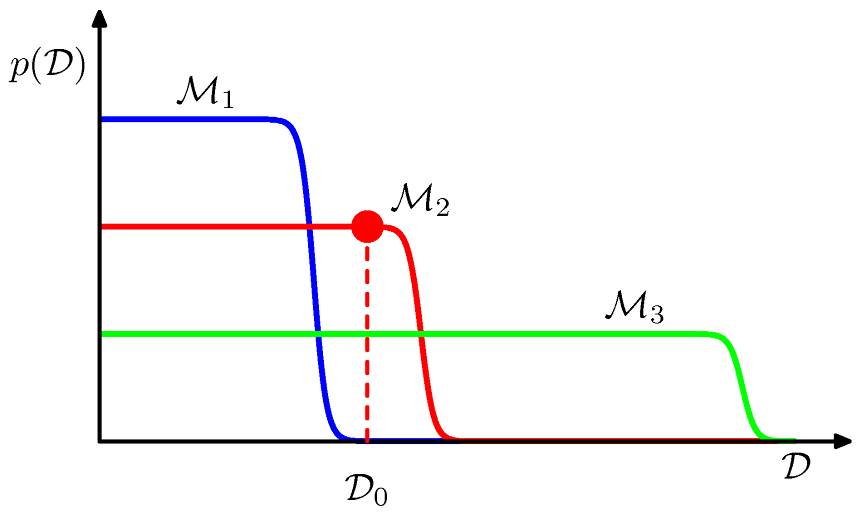
Bishop (2006) sec.3.4

Bayesian model comparison

- What can we do with model evidence?

Posterior over model space:

$$p(\mathcal{M}_i \mid \mathcal{D}) \propto p(\mathcal{M}_i)\, p(\mathcal{D} \mid \mathcal{M}_i)$$



Bayesian model comparison

- What can we do with model evidence?

Posterior over model space: $$p(\mathcal{M}_i \mid \mathcal{D}) \propto p(\mathcal{M}_i)\, p(\mathcal{D} \mid \mathcal{M}_i)$$

- prediction can be made using the "best" model (single most probable model) or can be averaged from the mixture of K models

Model mixture: $$p(t \mid \mathbf{x}, \mathcal{D}) = \sum_{i=1}^{K} p(t \mid \mathbf{x}, \mathcal{M}_i, \mathcal{D})\, p(\mathcal{M}_i \mid \mathcal{D})$$

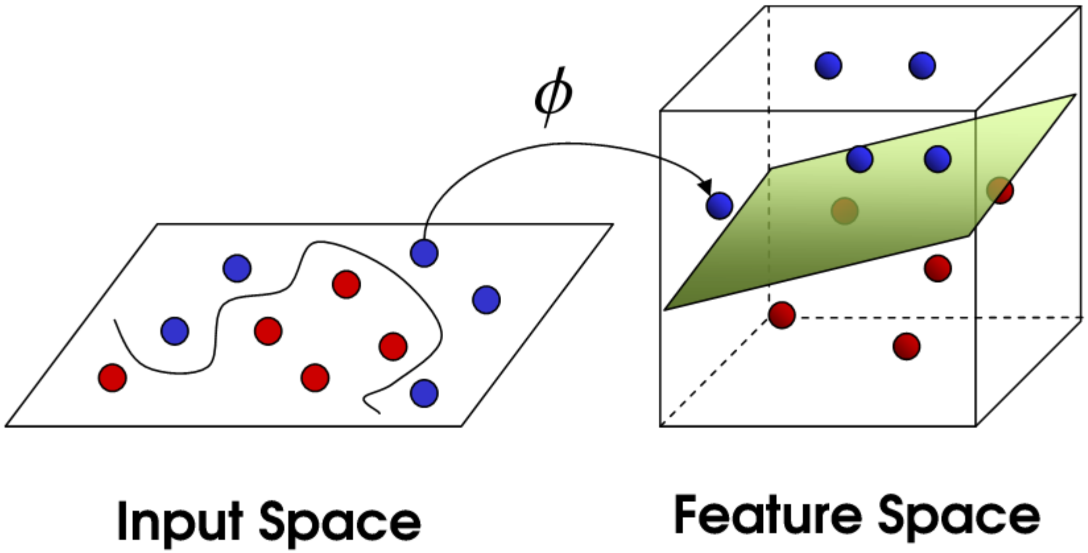
Holistic Bayesian approach

i. Choose a model
ii. Formulate prediction error by likelihood
iii. Formulate belief of a model in prior
iv. Marginalise irrelevant variables (parameters)
v. Choose model based on evidence
vi. Make predictions

Prediction should reflect my beliefs in the model and the information in the observations.

Kernels

- Dual linear regression
- Dual representations
- Kernel functions



Philipp Wagner: *Machine Learning with OpenCV2*

Dual linear regression

$$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}) \propto p(\mathbf{t} \mid \mathbf{w}, \mathbf{X})\ p(\mathbf{w})$$

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N), \quad \mathbf{t} = (t_1, \ldots, t_N)$$

$$p(\mathbf{t} \mid \mathbf{w}, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid \mathbf{w}^{\mathrm{T}} \mathbf{x}_n, \sigma^{-1}\mathbf{I}\right)$$

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} \mid 0, \tau^{-1}\mathbf{I}\right)$$

Dual linear regression

$$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}) \propto p(\mathbf{t} \mid \mathbf{w}, \mathbf{X})\, p(\mathbf{w})$$

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N), \qquad \mathbf{t} = (t_1, \ldots, t_N)$$

$$p(\mathbf{t} \mid \mathbf{w}, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid \mathbf{w}^{\mathrm{T}} \mathbf{x}_n, \sigma^{-1} \mathbf{I}\right)$$

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w} \mid 0, \tau^{-1} \mathbf{I}\right)$$

$$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}) \propto \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{1}{2\tau^2}\left(\mathbf{w}^{\mathrm{T}}\mathbf{w}\right)} \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(t_n - \mathbf{w}^{\mathrm{T}}\mathbf{x}_n\right)^{\mathrm{T}}\left(t_n - \mathbf{w}^{\mathrm{T}}\mathbf{x}_n\right)}$$

$$= \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{1}{2\tau^2}\left(\mathbf{w}^{\mathrm{T}}\mathbf{w}\right)} \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} e^{-\frac{1}{2\sigma^2}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right)^{\mathrm{T}}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right)}$$

Dual linear regression

For this linear (in parameters $\mathbf{w}$ and inputs $\mathbf{x}$) regression problem, let's find the maximum posterior solution, i.e.

$$\arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

Dual linear regression

For this linear (in parameters $\mathbf{w}$ and inputs $\mathbf{x}$) regression problem, let's find the maximum posterior solution, i.e.

$$\arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\Downarrow \quad L(\mathbf{w}) \propto \ln p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\mathbf{w}} \left\{ \frac{1}{2}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right)^{\mathrm{T}}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \right\}$$

Dual linear regression

For this linear (in parameters $\mathbf{w}$ and inputs $\mathbf{x}$) regression problem, let's find the maximum posterior solution, i.e.

$$\arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\Downarrow \quad L(\mathbf{w}) \propto \ln p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\mathbf{w}} \left\{ \frac{1}{2} \left(\mathbf{t} - \mathbf{w}^{\mathrm{T}} \mathbf{X} \right)^{\mathrm{T}} \left(\mathbf{t} - \mathbf{w}^{\mathrm{T}} \mathbf{X} \right) + \frac{\lambda}{2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \right\}$$

$$\Downarrow \quad \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 0$$

$$\mathbf{w} = -\frac{1}{\lambda} \mathbf{X}^{\mathrm{T}} \left(\mathbf{w}^{\mathrm{T}} \mathbf{X} - \mathbf{t} \right) = \mathbf{X}^{\mathrm{T}} \mathbf{a} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n$$

Dual linear regression

For this linear (in parameters $\mathbf{w}$ and inputs $\mathbf{x}$) regression problem, let's find the maximum posterior solution, i.e.

$$\arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\Downarrow \quad L(\mathbf{w}) \propto \ln p(\mathbf{w} \mid \mathbf{t}, \mathbf{X})$$

$$\arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\mathbf{w}} \left\{ \frac{1}{2}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right)^{\mathrm{T}}\left(\mathbf{t} - \mathbf{w}^{\mathrm{T}}\mathbf{X}\right) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \right\}$$

$$\Downarrow \quad \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 0$$

$$\mathbf{w} = -\frac{1}{\lambda}\mathbf{X}^{\mathrm{T}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{X} - \mathbf{t}\right) = \mathbf{X}^{\mathrm{T}}\mathbf{a} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n$$

$$\boxed{\alpha_n = -\frac{1}{\lambda}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n - \mathbf{t}_n\right)}$$

Dual linear regression

Let's define a *dual representation* of L by $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a} = \sum_{n}^{N} \alpha_n \mathbf{x}_n$

$$L(\mathbf{w}) \rightarrow L(\mathbf{a}): \quad L(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

Dual linear regression

Let's define a *dual representation* of L by $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a} = \sum_{n}^{N} \alpha_n \mathbf{x}_n$

$$L(\mathbf{w}) \rightarrow L(\mathbf{a}): \quad L(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

If we define a symmetric matrix $\mathbf{K}$:

$$[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$

$k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function that measures here the inner product between $\mathbf{x}_i$ and $\mathbf{x}_j$

Dual linear regression

Let's define a *dual representation* of L by $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{a} = \sum_{n}^{N} \alpha_n \mathbf{x}_n$

$$L(\mathbf{w}) \rightarrow L(\mathbf{a}): \quad L(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

If we define a symmetric matrix $\mathbf{K}$:

$$[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$

$$\Downarrow$$

$$L(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} \quad \overset{\frac{\partial L(\mathbf{a})}{\partial \mathbf{a}} = 0}{\Longrightarrow} \quad \mathbf{a} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}$$

Dual linear regression

So, for a new input x^*, the model generates prediction t^*:

$$t^* = \mathbf{w}^{\mathrm{T}}\mathbf{x}^* = \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{x}^* = \mathbf{a}^{\mathrm{T}}\boldsymbol{k}(\mathbf{x}^*) =$$

$$= \left((\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{t}\right)^{\mathrm{T}}\boldsymbol{k}(\mathbf{x}^*) = \boldsymbol{k}(\mathbf{x}^*)^{\mathrm{T}}(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{t}$$

$$\boldsymbol{k}(\mathbf{x}^*) = \left(k_1(\mathbf{x}^*),\ldots,k_N(\mathbf{x}^*)\right)^{\mathrm{T}}, \qquad k_n(\mathbf{x}^*) = k\left(\mathbf{x}_n,\mathbf{x}^*\right)$$

Dual linear regression

So, for a new input x^*, the model generates prediction t^*:

$$t^* = \mathbf{w}^{\mathrm{T}}\mathbf{x}^* = \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{x}^* = \mathbf{a}^{\mathrm{T}}\boldsymbol{k}(\mathbf{x}^*) =$$

$$= \left((\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t} \right)^{\mathrm{T}} \boldsymbol{k}(\mathbf{x}^*) = \boldsymbol{k}(\mathbf{x}^*)^{\mathrm{T}} (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}$$

$$\boldsymbol{k}(\mathbf{x}^*) = \left(k_1(\mathbf{x}^*), \ldots, k_N(\mathbf{x}^*) \right)^{\mathrm{T}}, \qquad k_n(\mathbf{x}^*) = k\left(\mathbf{x}_n, \mathbf{x}^*\right)$$

$$t = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \qquad \boldsymbol{\phi}(\mathbf{x}) = \left(\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}) \right)^{\mathrm{T}}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_j), \qquad \mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}, \qquad \boldsymbol{\Phi} = \left(\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N) \right)$$

$$t^* = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}^*) = \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\,\boldsymbol{\phi}(\mathbf{x}^*) = \boldsymbol{k}(\mathbf{x}^*)^{\mathrm{T}} (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}$$

Dual linear regression

- Linear regression

  i. Use training data $\mathcal{D}_{trn} = (\mathbf{x}, t)_i^N$ to learn and encode relationship in $\mathbf{w}$.

  ii. Throw away training data.

  iii. Make predictions for new test data using model defined by $\mathbf{w}$.

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Dual linear regression

- Linear regression

  i. Use training data $\mathcal{D}_{trn} = (\mathbf{x},t)_i^N$ to learn and encode relationship in $\mathbf{w}$.

  ii. Throw away training data.

  iii. Make predictions for new test data using model defined by $\mathbf{w}$.

- Dual regression

  i. Use training data and measure pair-wise distances.

  ii. Do NOT throw away training data even for prediction.

  iii. Make predictions using distance/similarity relationship to training data.

  iv. Model complexity depends on data (it adapts).

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels and feature mapping

For regression linear in the inputs $\boldsymbol{x}$

$$t_{out} = \boldsymbol{k}(\mathbf{x}_{in})^{\mathrm{T}} \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{t}, \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\mathrm{T}}, \quad \boldsymbol{k}(\mathbf{x}_{in}) = \left(\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_{in}, \ldots, \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_{in}\right)^{\mathrm{T}}$$

and for the input mapped to some other space, $\boldsymbol{\phi}(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^M$

$$t_{out} = \boldsymbol{k}(\mathbf{x}_{in})^{\mathrm{T}} \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{t}, \quad \mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}},$$

$$\boldsymbol{k}(\mathbf{x}_{in}) = \left(\underbrace{\boldsymbol{\phi}(\mathbf{x}_1)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{in})}, \ldots, \underbrace{\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{in})}\right)^{\mathrm{T}}$$

Kernels and feature mapping

For regression linear in the inputs $\boldsymbol{x}$

and for the input mapped to some other space,

$$t_{out} = \boldsymbol{k}(\mathbf{x}_{in})^{\mathrm{T}} \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{t}, \quad \mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}},$$

$$\boldsymbol{k}(\mathbf{x}_{in}) = \left(\underbrace{\phi(\mathbf{x}_1)^{\mathrm{T}} \phi(\mathbf{x}_{in})}, \ldots, \underbrace{\phi(\mathbf{x}_n)^{\mathrm{T}} \phi(\mathbf{x}_{in})}\right)^{\mathrm{T}}$$

the calculations rely on inner products of the data.

Kernels and feature mapping

For regression linear in the inputs $\boldsymbol{x}$

$$t_{out} = \boldsymbol{k}(\mathbf{x}_{in})^{\mathrm{T}} \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{t}, \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\mathrm{T}}, \quad \boldsymbol{k}(\mathbf{x}_{in}) = \left(\mathbf{x}_1^{\mathrm{T}}\mathbf{x}_{in}, \ldots, \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_{in}\right)^{\mathrm{T}}$$

and for the input mapped to some other space, $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^M$

$$t_{out} = \boldsymbol{k}(\mathbf{x}_{in})^{\mathrm{T}} \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{t}, \quad \mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}},$$

$$\boldsymbol{k}(\mathbf{x}_{in}) = \left(\underbrace{\boldsymbol{\phi}(\mathbf{x}_1)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{in})}, \ldots, \underbrace{\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_{in})}\right)^{\mathrm{T}}$$

the calculations rely on inner products of the data.

Conclusion: we do not need to know $\boldsymbol{\phi}(\cdot)$, only $\boldsymbol{\phi}(\cdot)^{\mathrm{T}} \boldsymbol{\phi}(\cdot)$.

Kernel functions

Kernel function $k(\mathbf{x}_i, \mathbf{x}_k)$ describes the inner product (distance)

$$k(\mathbf{x}_i, \mathbf{x}_k) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) = \|\phi(\mathbf{x}_i)\| \|\phi(\mathbf{x}_k)\| \cos(\theta)$$

Kernel functions

Kernel function $k(\mathbf{x}_i, \mathbf{x}_k)$ describes the inner product (similarity)

$$k(\mathbf{x}_i, \mathbf{x}_k) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) = \|\phi(\mathbf{x}_i)\| \|\phi(\mathbf{x}_k)\| \cos(\theta)$$

- Kernels are subclass of functions:

  K matrix (the Gram matrix) must be *positive semidefinite*

Kernel functions

Kernel function $k(\mathbf{x}_i, \mathbf{x}_k)$ describes the inner product (similarity)

$$k(\mathbf{x}_i, \mathbf{x}_k) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) = \|\phi(\mathbf{x}_i)\| \|\phi(\mathbf{x}_k)\| \cos(\theta)$$

- Kernels are subclass of functions:

  K matrix (the Gram matrix) must be *positive semidefinite*

- Examples:

$$k(\mathbf{x}_i, \mathbf{x}_k) = \exp(-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}_k\|^2)$$

$$k(\mathbf{x}_i, \mathbf{x}_k) = \left(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_k + c\right)^M$$

Kernel functions

Kernel function $k(\mathbf{x}_i, \mathbf{x}_k)$ describes the inner product (similarity)

$$k(\mathbf{x}_i, \mathbf{x}_k) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) = \|\phi(\mathbf{x}_i)\| \|\phi(\mathbf{x}_k)\| \cos(\theta)$$

- Techniques to construct kernels from existing kernels (Bishop, p.296)

$$k(\mathbf{x}_i, \mathbf{x}_k) = k_1(\mathbf{x}_i, \mathbf{x}_k) + k_2(\mathbf{x}_i, \mathbf{x}_k)$$

$$k(\mathbf{x}_i, \mathbf{x}_k) = k_1(\mathbf{x}_i, \mathbf{x}_k) \cdot k_2(\mathbf{x}_i, \mathbf{x}_k)$$

$$k(\mathbf{x}_i, \mathbf{x}_k) = k_a(\mathbf{x}_i^{(a)}, \mathbf{x}_k^{(a)}) + k_b(\mathbf{x}_i^{(b)}, \mathbf{x}_k^{(b)})$$

$$k(\mathbf{x}_i, \mathbf{x}_k) = k_a(\mathbf{x}_i^{(a)}, \mathbf{x}_k^{(a)}) \cdot k_b(\mathbf{x}_i^{(b)}, \mathbf{x}_k^{(b)})$$

Kernel functions

Kernel function $k(\mathbf{x}_i, \mathbf{x}_k)$ describes the inner product (similarity)

$$k(\mathbf{x}_i,\mathbf{x}_k) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) = \left\|\phi(\mathbf{x}_i)\right\| \left\|\phi(\mathbf{x}_k)\right\| \cos(\theta)$$

- Techniques to construct kernels from existing kernels (Bishop, p.296)

$$k(\mathbf{x}_i,\mathbf{x}_k) = k_1(\mathbf{x}_i,\mathbf{x}_k) + k_2(\mathbf{x}_i,\mathbf{x}_k)$$
$$k(\mathbf{x}_i,\mathbf{x}_k) = k_1(\mathbf{x}_i,\mathbf{x}_k) \cdot k_2(\mathbf{x}_i,\mathbf{x}_k)$$
$$k(\mathbf{x}_i,\mathbf{x}_k) = k_a(\mathbf{x}_i^{(a)},\mathbf{x}_k^{(a)}) + k_b(\mathbf{x}_i^{(b)},\mathbf{x}_k^{(b)})$$
$$k(\mathbf{x}_i,\mathbf{x}_k) = k_a(\mathbf{x}_i^{(a)},\mathbf{x}_k^{(a)}) \cdot k_b(\mathbf{x}_i^{(b)},\mathbf{x}_k^{(b)})$$

- With the kernel, we do not need to know the mapping $\boldsymbol{\phi}(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^M$

Kernel functions

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^{T}\mathbf{x}_j\right)^2, \quad \mathbf{x} \in \mathbb{R}^2$$

$$\begin{aligned}
(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^2 &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 = \\
&= x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 = \\
&= (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2)(x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2)^{\mathrm{T}} = \\
&= \phi(\mathbf{x}_i)^{\mathrm{T}}\phi(\mathbf{x}_j)
\end{aligned}$$

So, $k(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j\right)^2$ is the kernel of the following mapping:

$$\phi(\mathbf{x}) = \left(\left(\mathbf{e}_1^{\mathrm{T}}\mathbf{x}\right)^2, \sqrt{2}\mathbf{e}_1^{\mathrm{T}}\mathbf{x}\mathbf{e}_2^{\mathrm{T}}\mathbf{x}, \left(\mathbf{e}_2^{\mathrm{T}}\mathbf{x}\right)^2\right), \quad \mathbf{e}_1 = (1,0)^{\mathrm{T}}, \mathbf{e}_2 = (0,1)^{\mathrm{T}}$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Bishop (2006), p.295

Kernels summary

Advantages of the *implicit* feature mapping with kernels

✓ the feature space itself does not need to be explicitly known

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels summary

Advantages of the *implicit* feature mapping with kernels

- ✓ the feature space itself does not need to be explicitly known
- ✓ the feature space can have infinite dimensionality

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels summary

Advantages of the *implicit* feature mapping with kernels

✓ the feature space itself does not need to be explicitly known

✓ the feature space can have infinite dimensionality

✓ the mapping can be non-linear but the problem is still linear

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels summary

Advantages of the *implicit* feature mapping with kernels

- ✓ the feature space itself does not need to be explicitly known
- ✓ the feature space can have infinite dimensionality
- ✓ the mapping can be non-linear but the problem is still linear
- ✓ kernels define covariance between data points

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels summary

Advantages of the *implicit* feature mapping with kernels

- ✓ the feature space itself does not need to be explicitly known
- ✓ the feature space can have infinite dimensionality
- ✓ the mapping can be non-linear but the problem is still linear
- ✓ kernels define covariance between data points
- ✓ there is even no need to work with vector data as long as it can be mapped to some vector space where scalar product can be computed, e.g. DNA strings

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Kernels summary

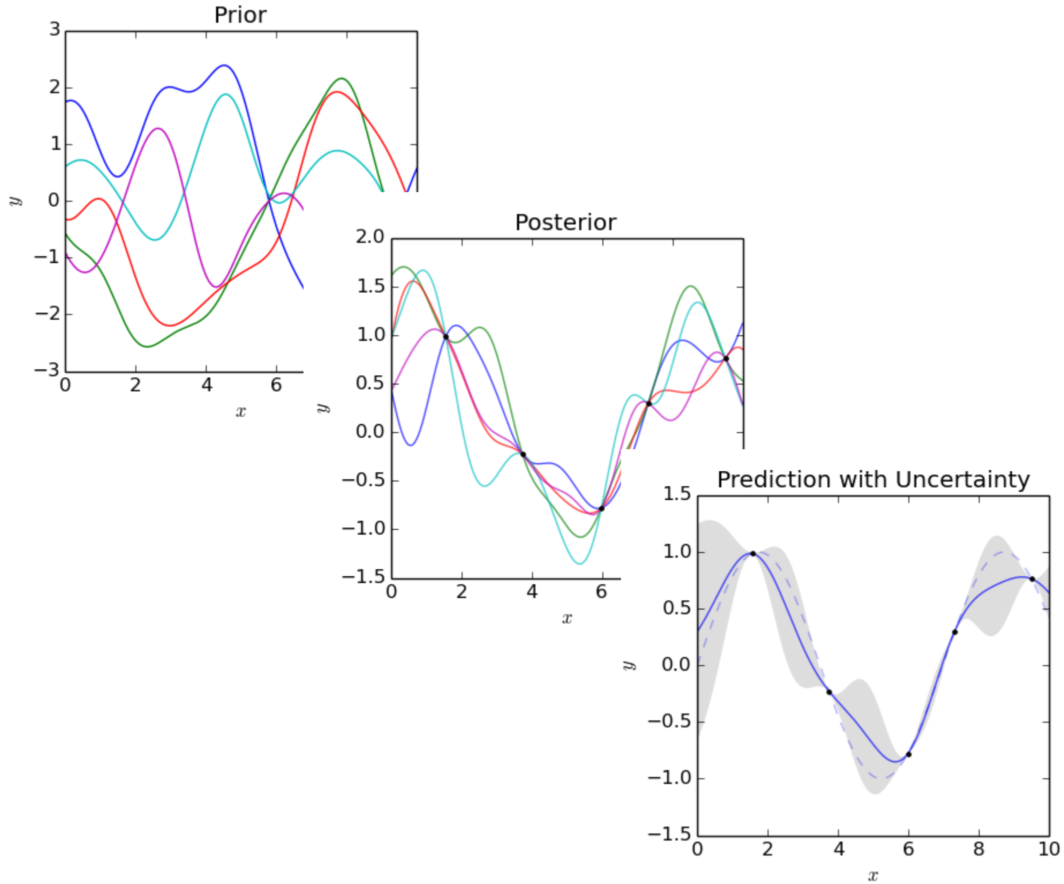
Advantages of the *implicit* feature mapping with kernels

- ✓ the feature space itself does not need to be explicitly known
- ✓ the feature space can have infinite dimensionality
- ✓ the mapping can be non-linear but the problem is still linear
- ✓ kern
- ✓ ther as it can be mapped to some vector space where scalar product can be computed, e.g. DNA strings

HOWEVER, for making predictions we need to keep all the training data even after training!

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, part I



Regression model

Model with noise

$$t_i = f(\mathbf{x}_i) + \varepsilon$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$

A new random variable f_i can be introduced

$$f_i = f(\mathbf{x}_i)$$

Instantiation of function f

Regression model

Joint distribution for the model

$$p(\mathbf{t},\mathbf{f},\mathbf{X},\theta) = p(\mathbf{t} \mid \mathbf{f})\, p(\boldsymbol{f} \mid \mathbf{X},\theta)\, p(\mathbf{X})\, p(\theta)$$

Prior over instantiations of function

$$p(\boldsymbol{f} \mid \mathbf{X},\theta)$$

It is the mapping f that we are uncertain of

Regression model

Joint distribution for the model

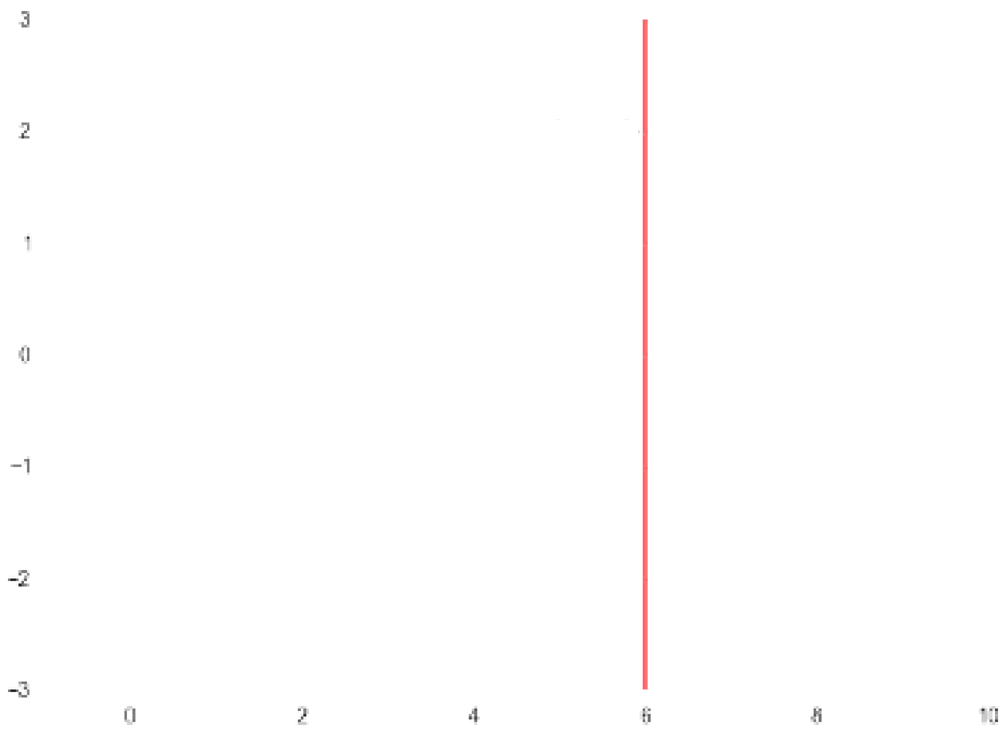
$$p(\mathbf{t},\mathbf{f},\mathbf{X},\theta) = p(\mathbf{t}\,|\,\mathbf{f})\,p(\boldsymbol{f}\,|\,\mathbf{X},\theta)\,p(\mathbf{X})\,p(\theta)$$

A new random variable f_i can be introduced

$$f_i = f(\mathbf{x}_i)$$

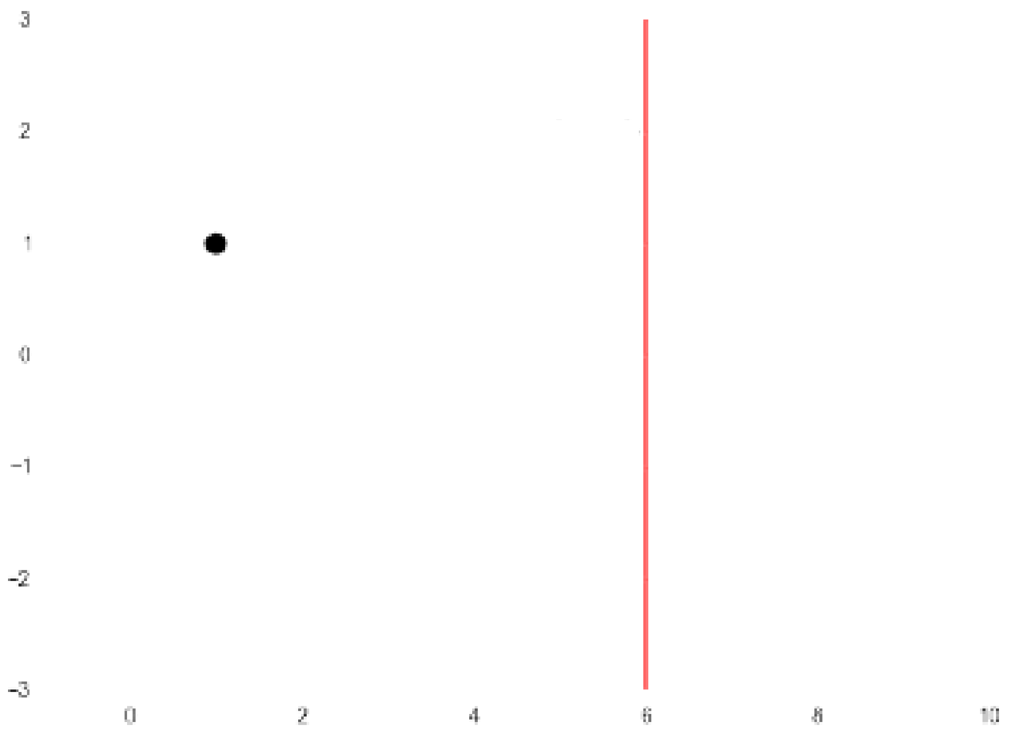
Instantiation of function f

Priors over functions



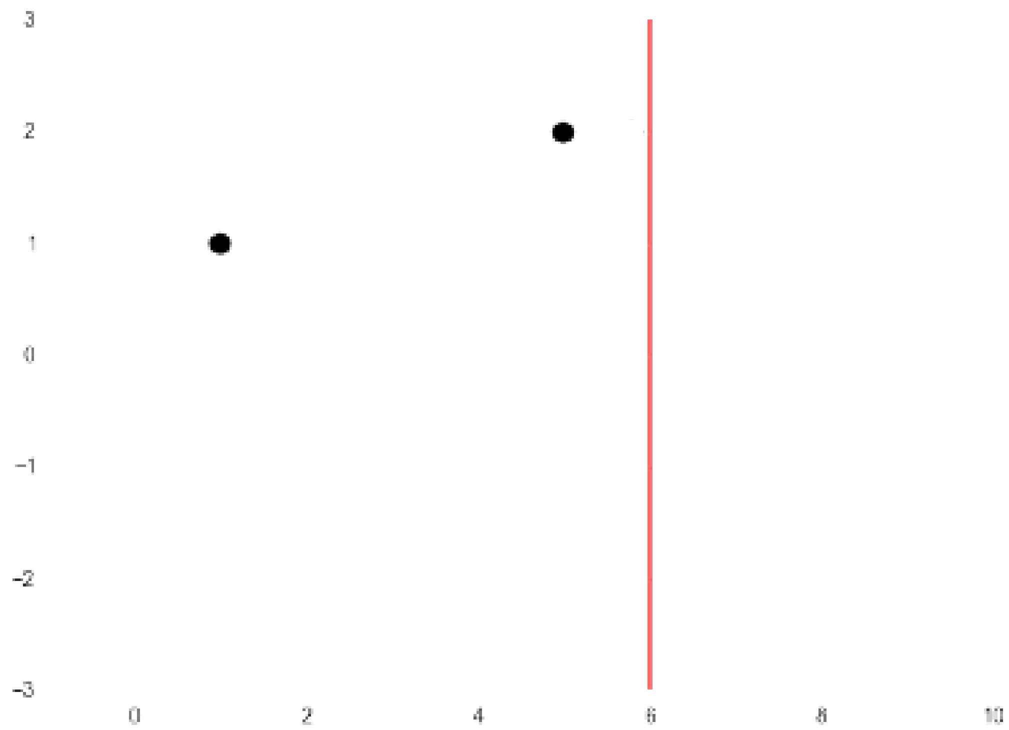
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Priors over functions



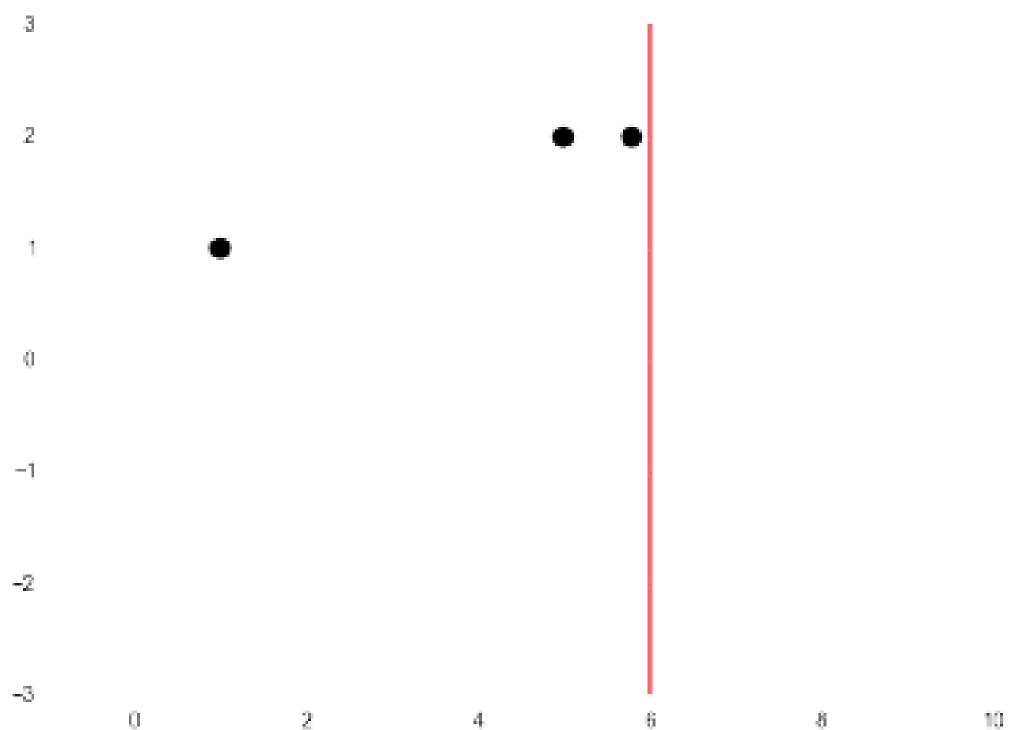
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Priors over functions



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Priors over functions



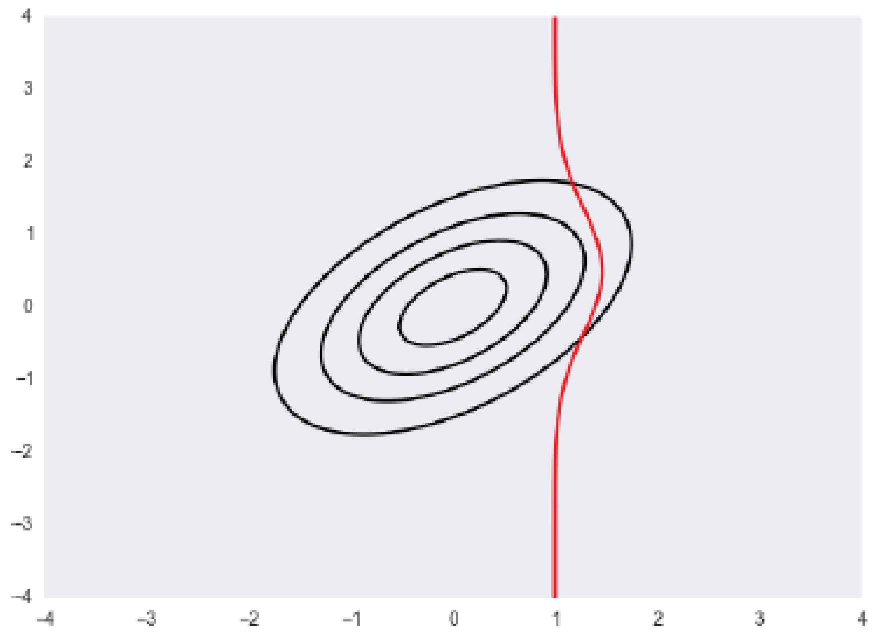
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

The Gaussian Conditional

$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

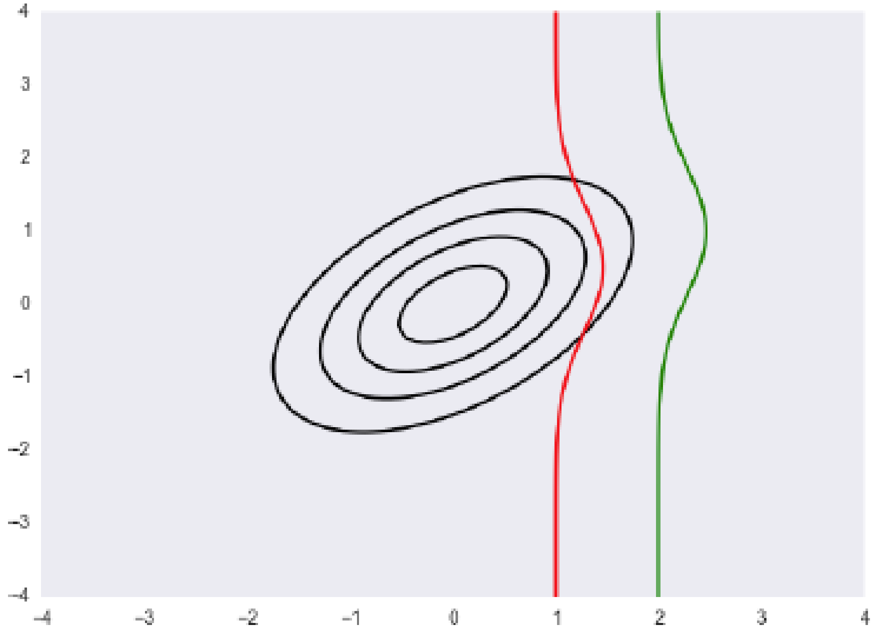
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

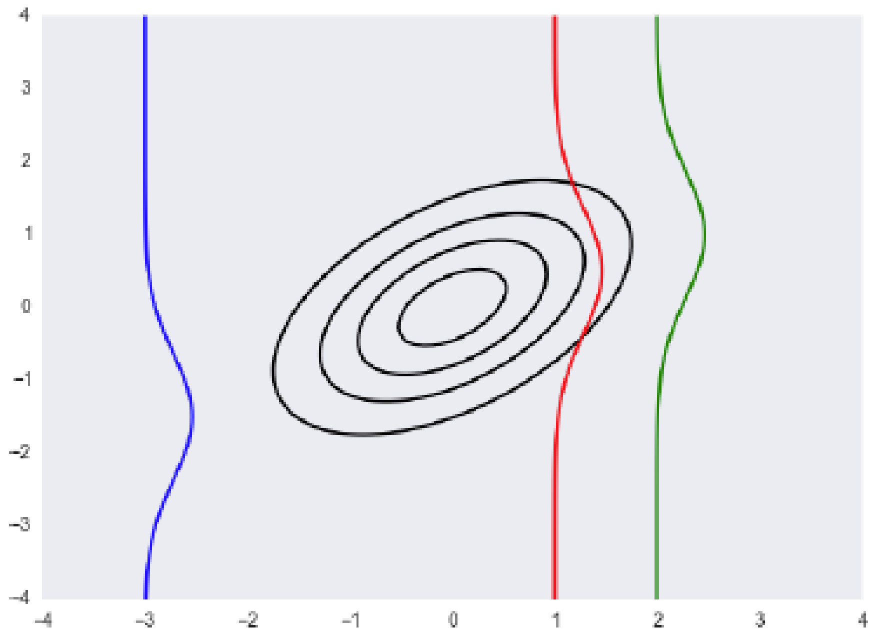
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

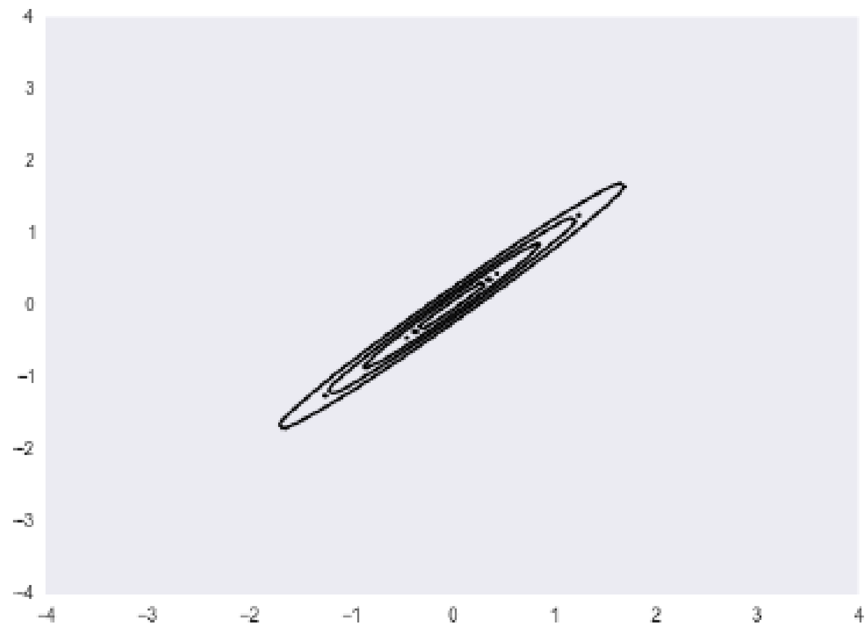
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

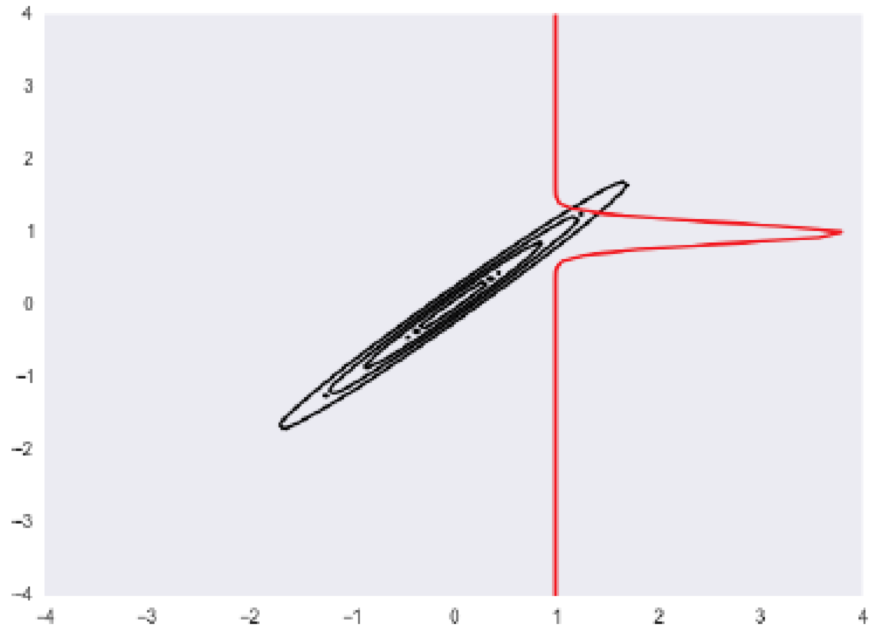
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.99\\0.99 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

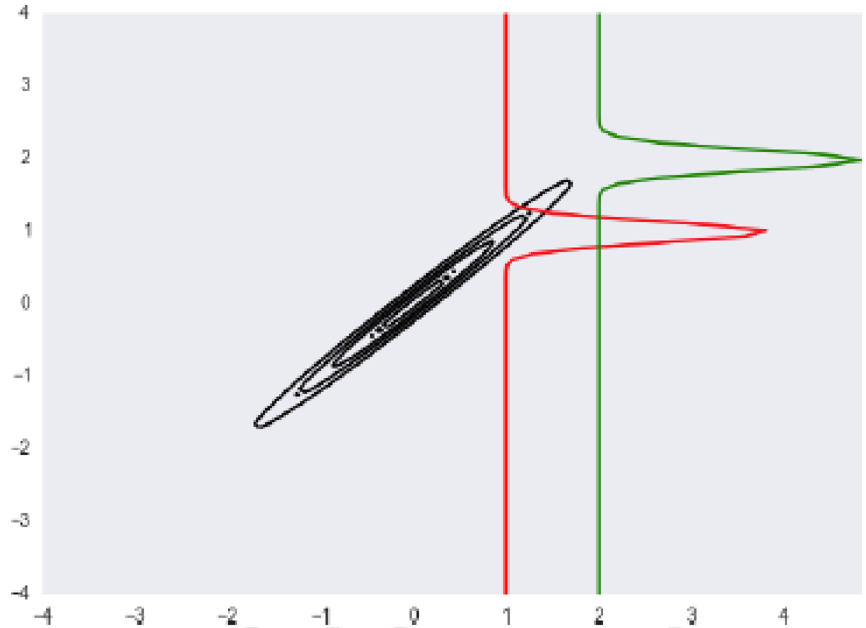
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.99\\0.99 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

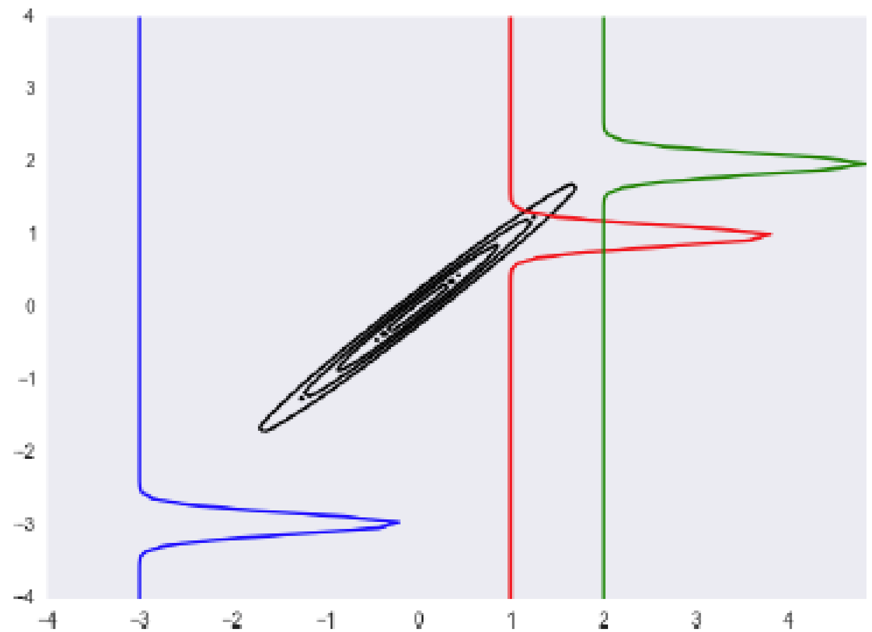
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.99\\0.99 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

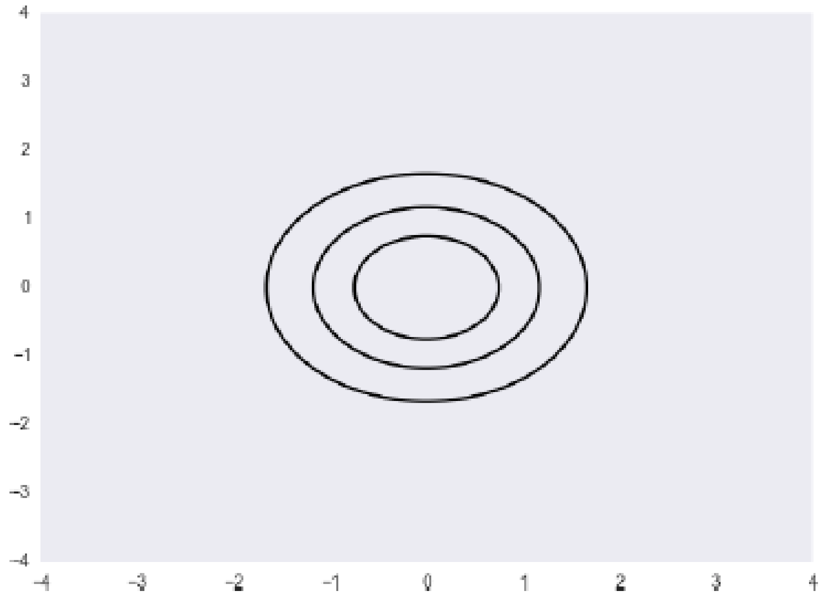
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0.99\\0.99 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

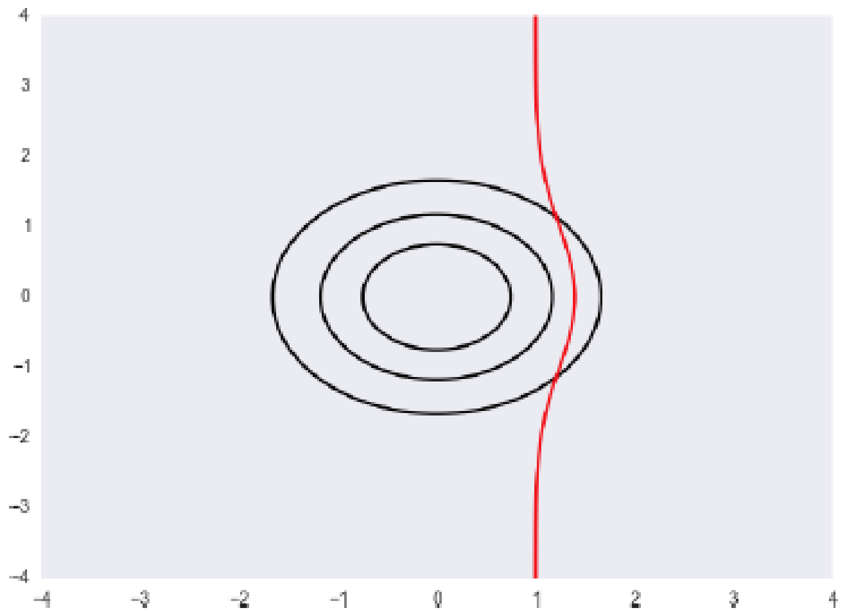
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

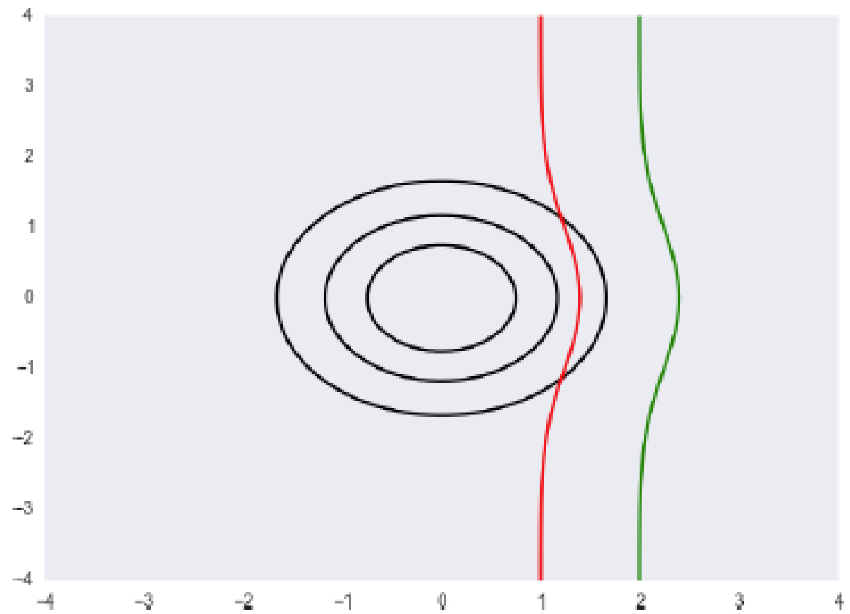
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

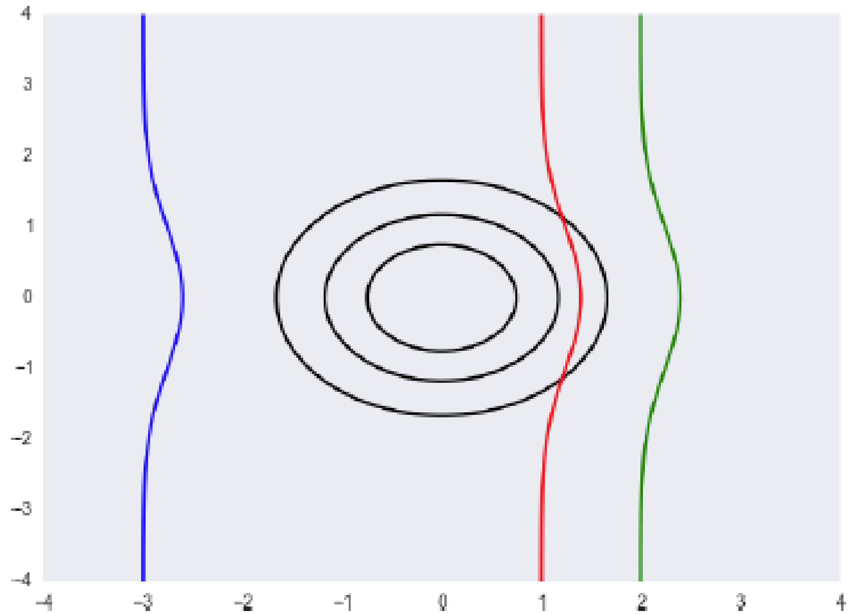
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

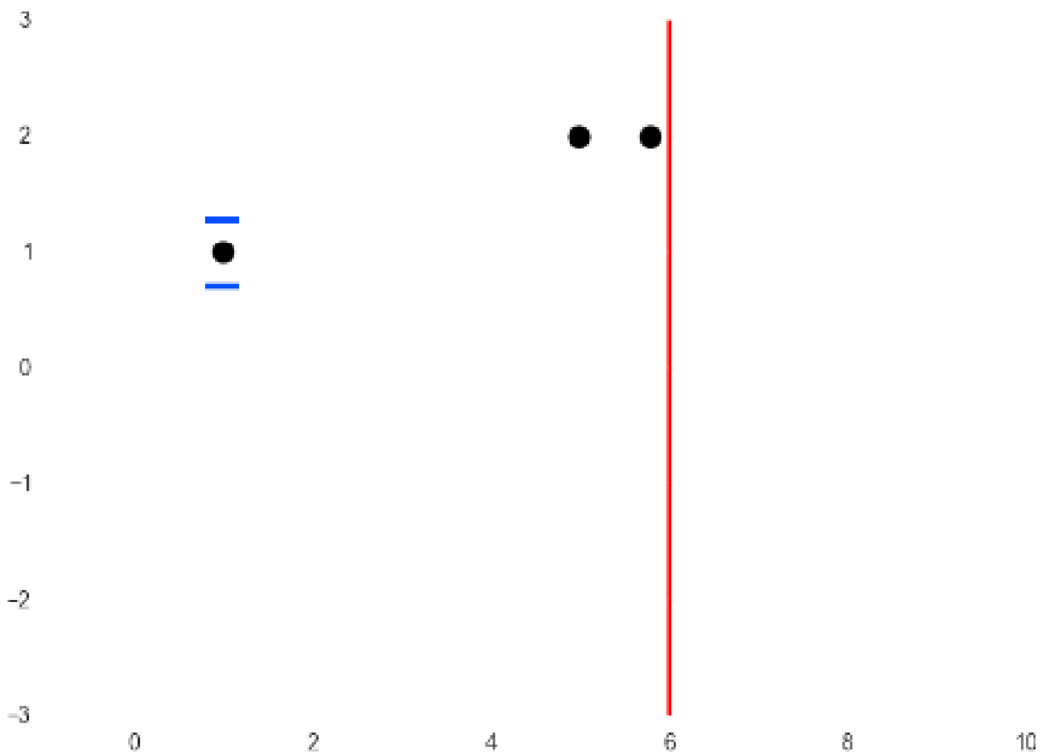
The Gaussian Conditional



$$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$$

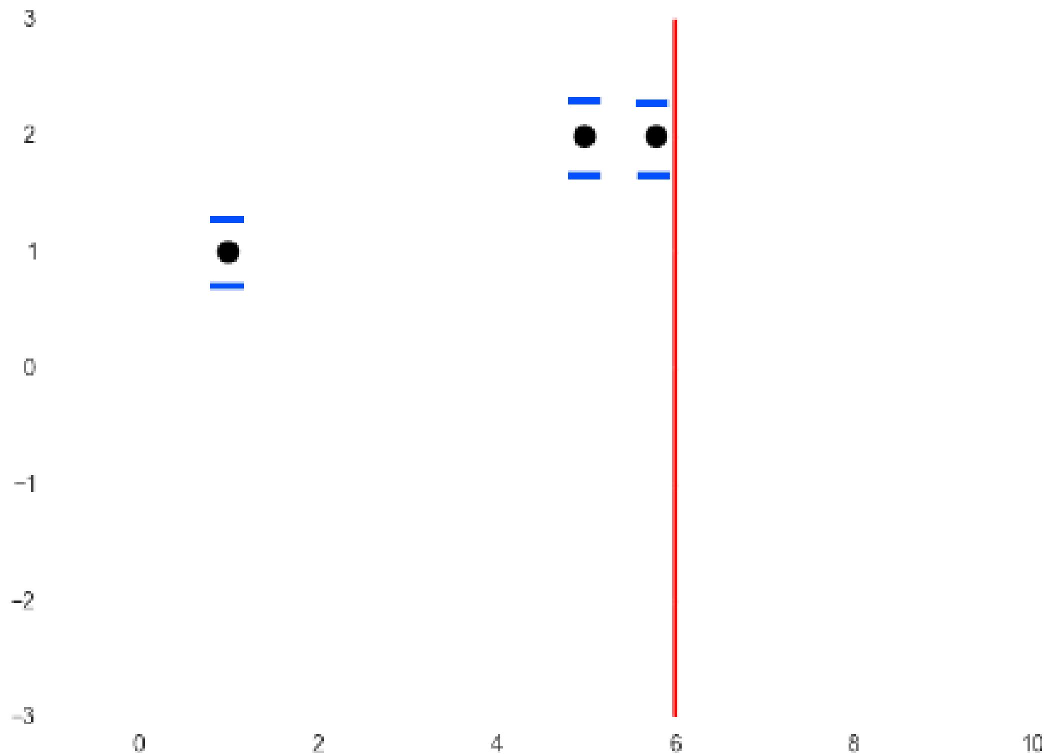
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuitive example



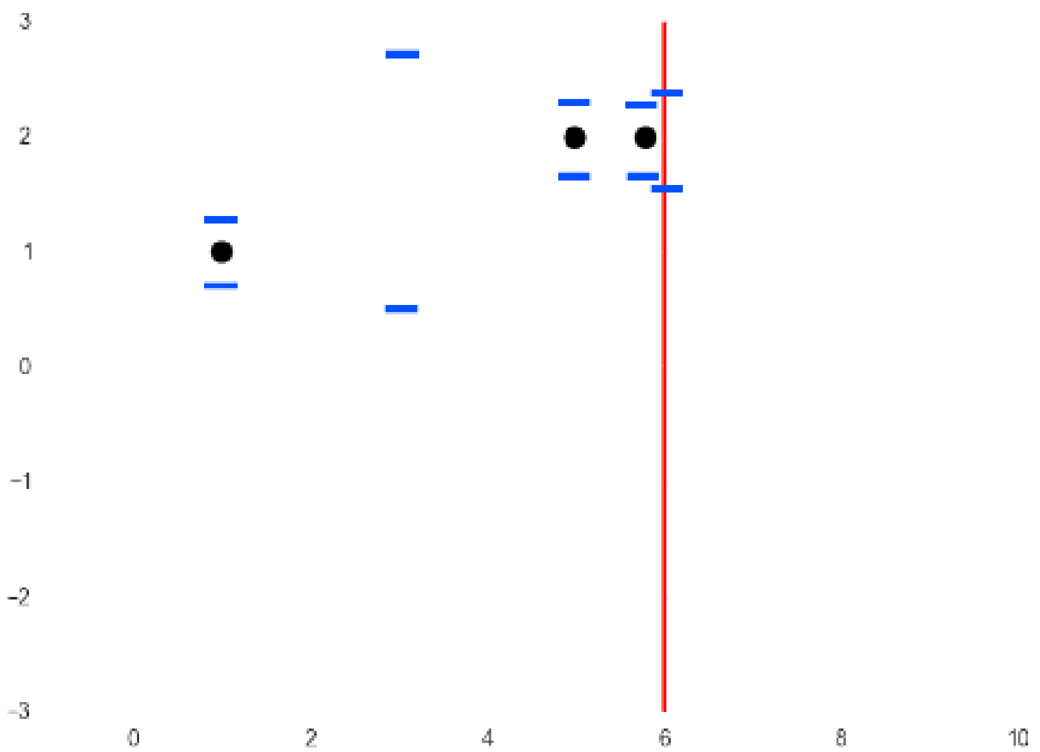
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuitive example



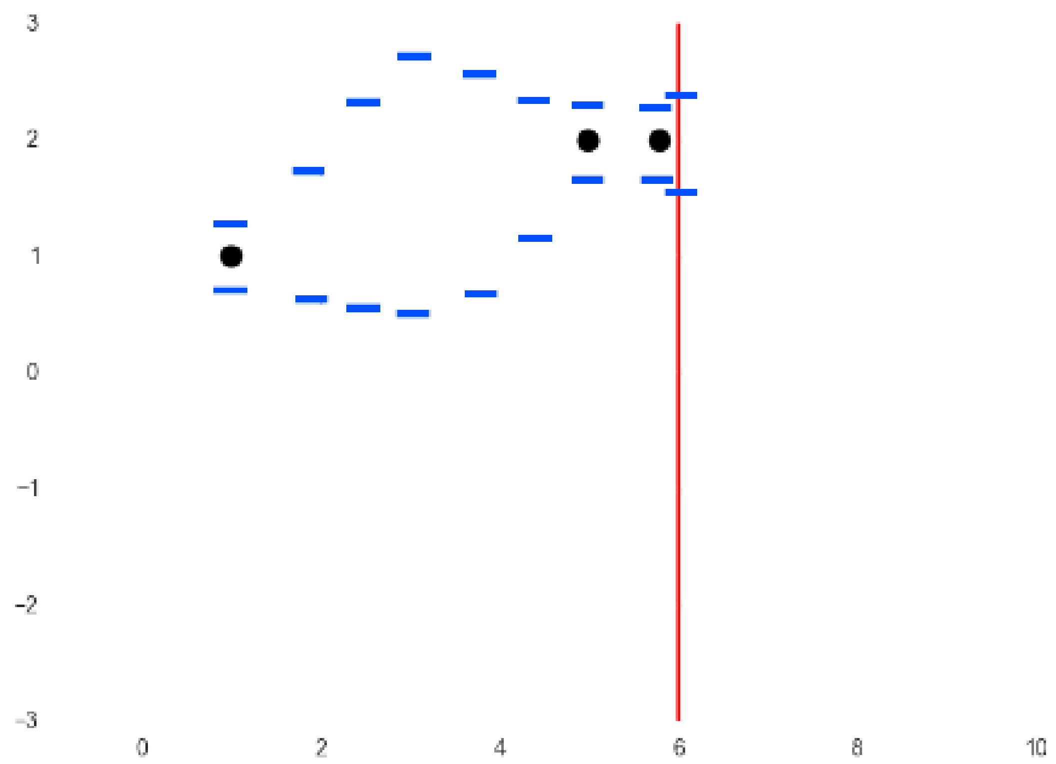
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuitive example



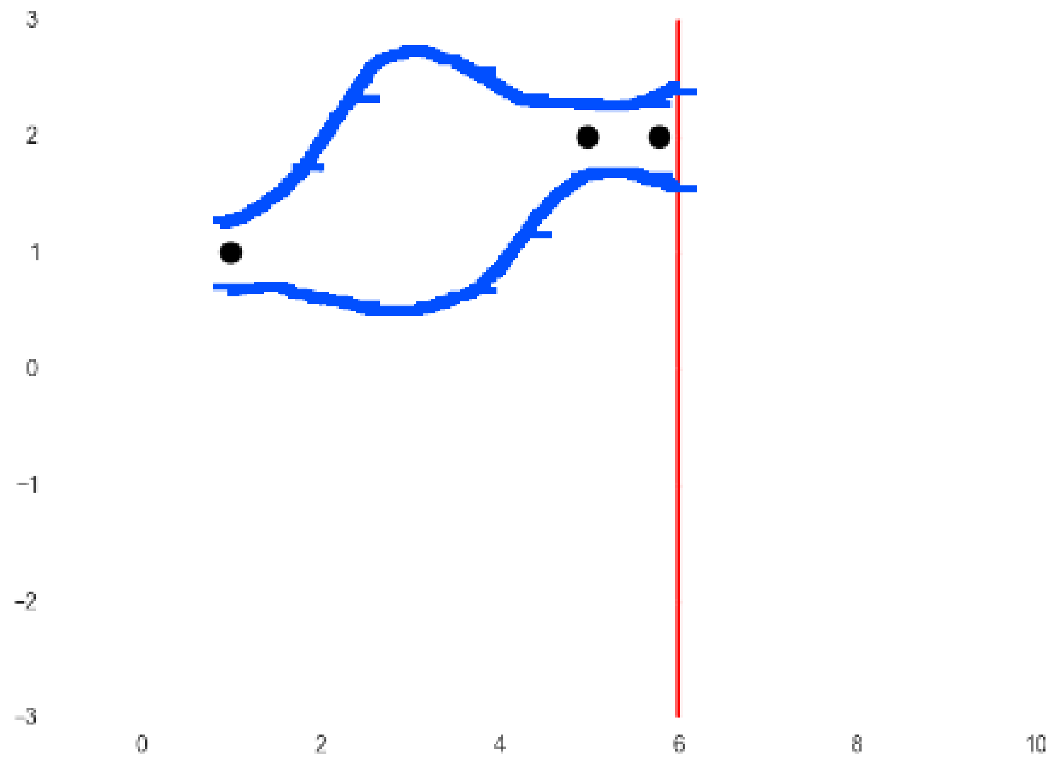
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuitive example



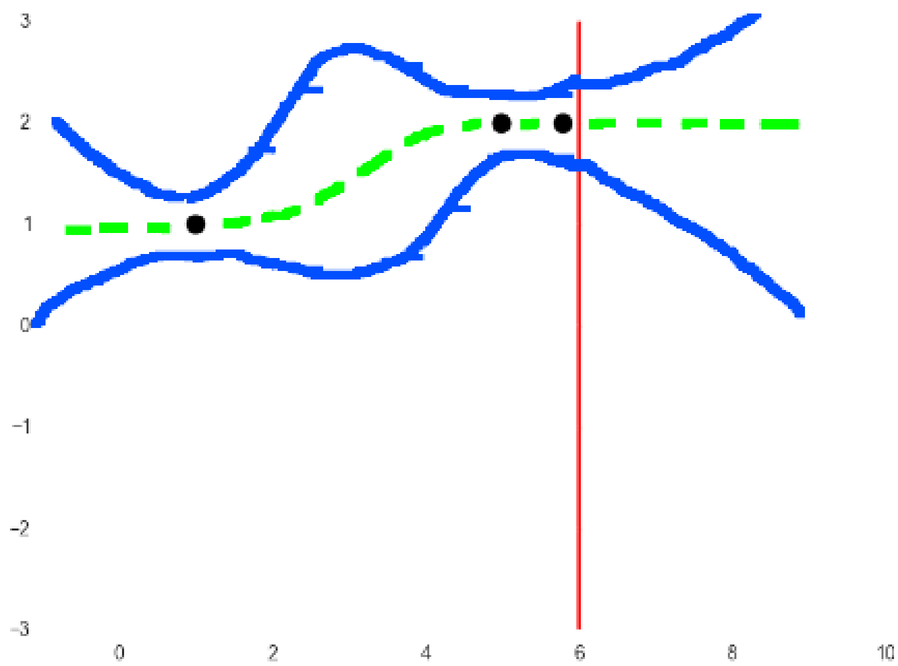
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuitive example



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

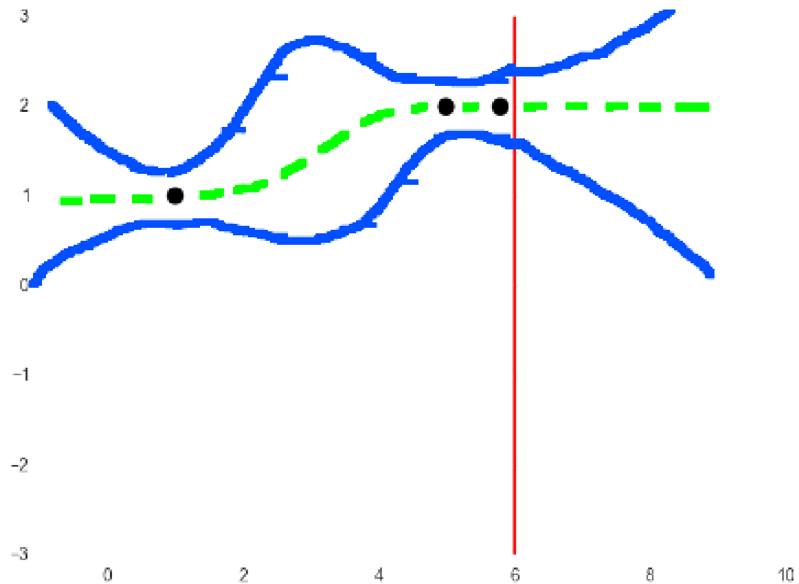
Intuition about the meaning of covariance



"If all instantiations of the function is jointly Gaussian such that the covariance structure depends on how much information an observation provides for the other we will get the curve above."

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Intuition about covariance



- Covariance between each point
- *Covariance function is a kernel*!
- All that can be done in the induced space (allow for any function)

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

What is a Gaussian process?

$$p(\boldsymbol{f} \mid \mathbf{X}, \theta) \sim \mathcal{GP}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X}))$$

A **Gaussian process (GP)** is an infinite collection of random variables so that any of their subsets is jointly Gaussian.

The process is specified by a mean function and covariance function:

$$f \sim \mathcal{GP}(\mu, k)$$

$$\mu = m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right],$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^{\mathrm{T}}\right]$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian process, model

$$p(\boldsymbol{f} \mid \mathbf{x}, \theta) = \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

$$\mathbf{t}_i = \boldsymbol{f}_i + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$

$$p(\mathbf{t} \mid \mathbf{X}, \theta) = \int p(\mathbf{t} \mid \boldsymbol{f})\, p(\boldsymbol{f} \mid \mathbf{X}, \theta)\, df$$

GP is infinite but only finite amount of data is observed, so conditioning on a data subset.

<u>GP is just a Gaussian distribution</u>, which is self-conjugate.

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Mean and covariance

- The mean function
  - Function of only the input location
  - The expectation of the function value only accounting for the input location

- The covariance function
  - Function of two input locations
  - The effect of information from other locations with known function value on my new estimate

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Mean and covariance

- The mean function
  - Function of only the input location
  - The expectation of the function value only accounting for the input location
  - It can be assumed to be constant

- The covariance function
  - Function of two input locations
  - The effect of information from other locations with known function value on my new estimate
  - Encodes behaviour of the function

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

GP example

The prior definition

$$p(\boldsymbol{f} \mid \mathbf{X}, \theta) = \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

$$\mu(\mathbf{x}) = 0$$

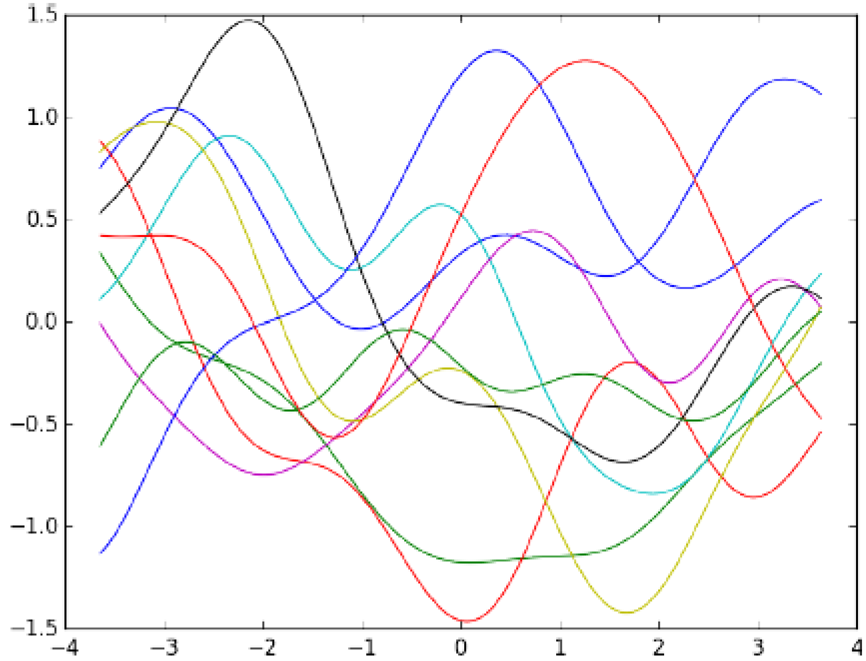
$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2\ell^2}\left(\mathbf{x}_i - \mathbf{x}_j\right)^{\mathrm{T}}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right)$$

squared exponential

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes

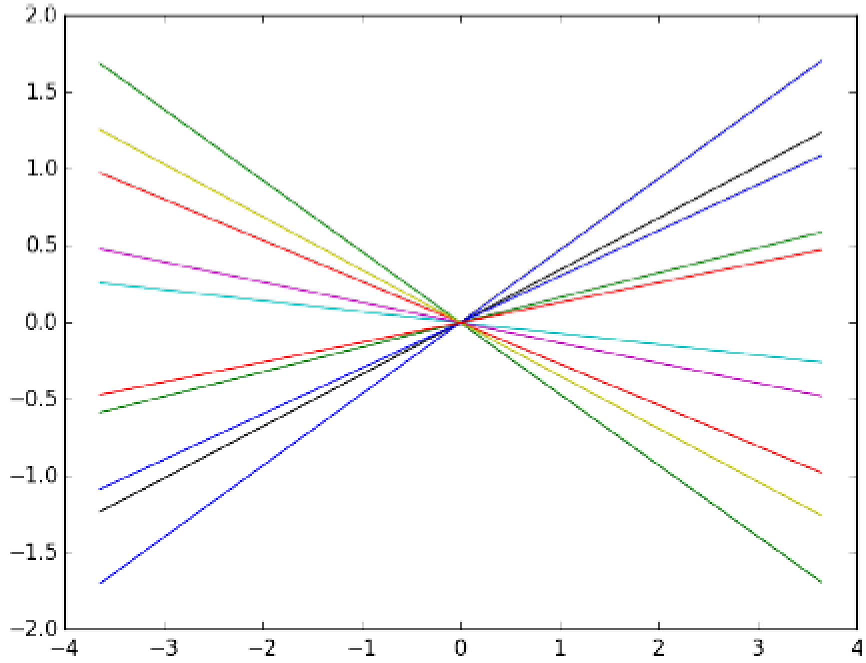
Let's draw a few samples for a given set of parameters (σ, ℓ)



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes

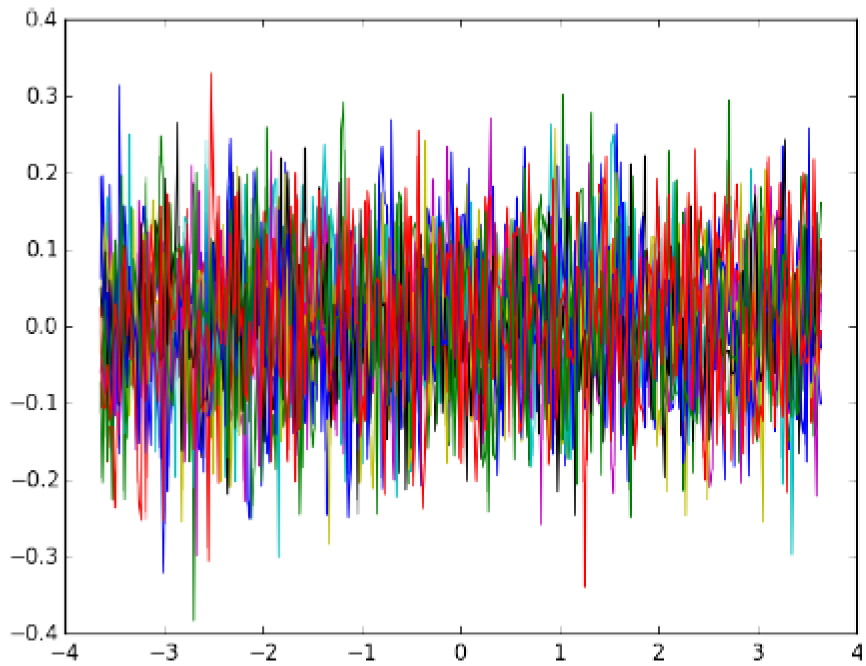
Linear kernel, $k(x_i, x_j) = \sigma^2 x_i x_j$



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

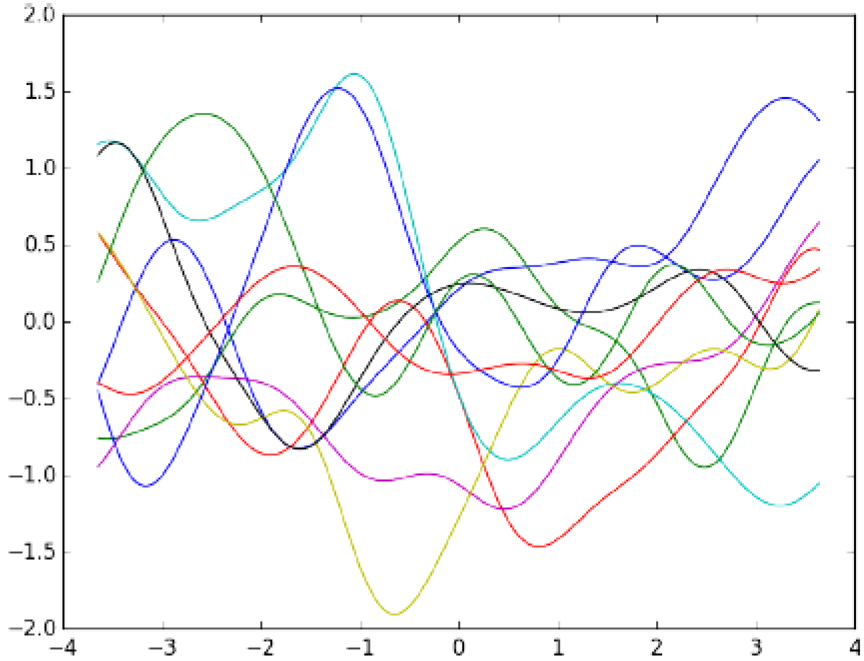
Samples from Gaussian processes

$$k(x_i, x_j) = 0$$



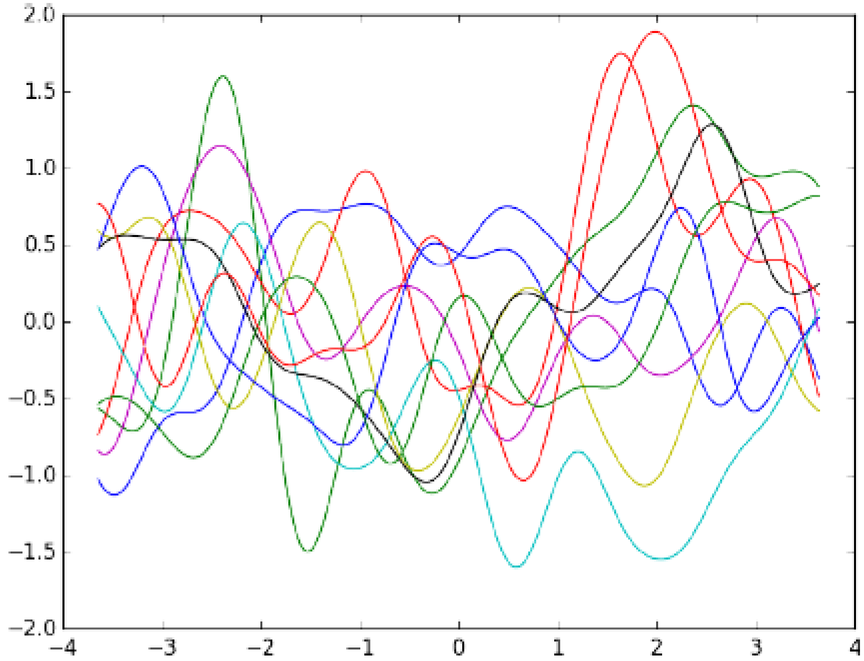
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



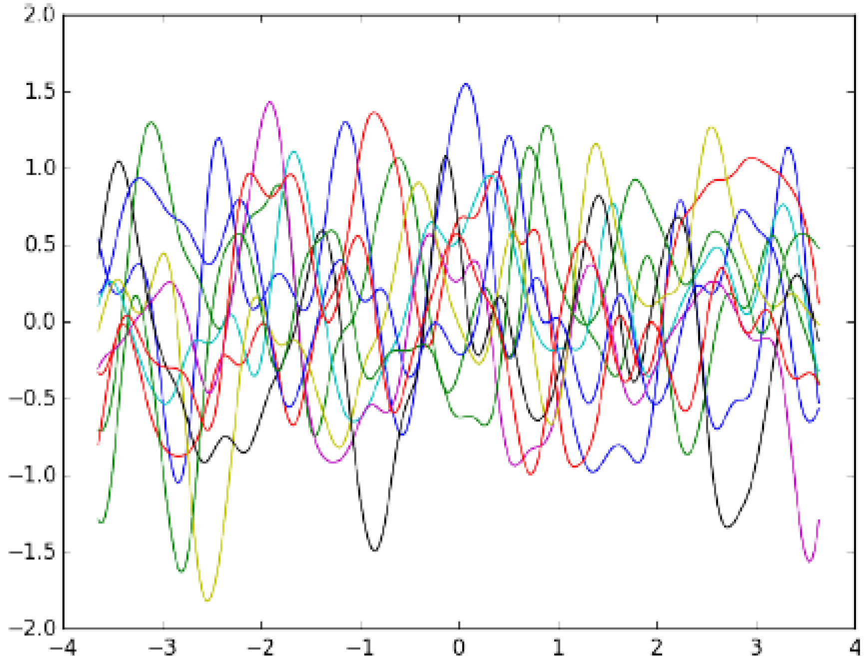
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



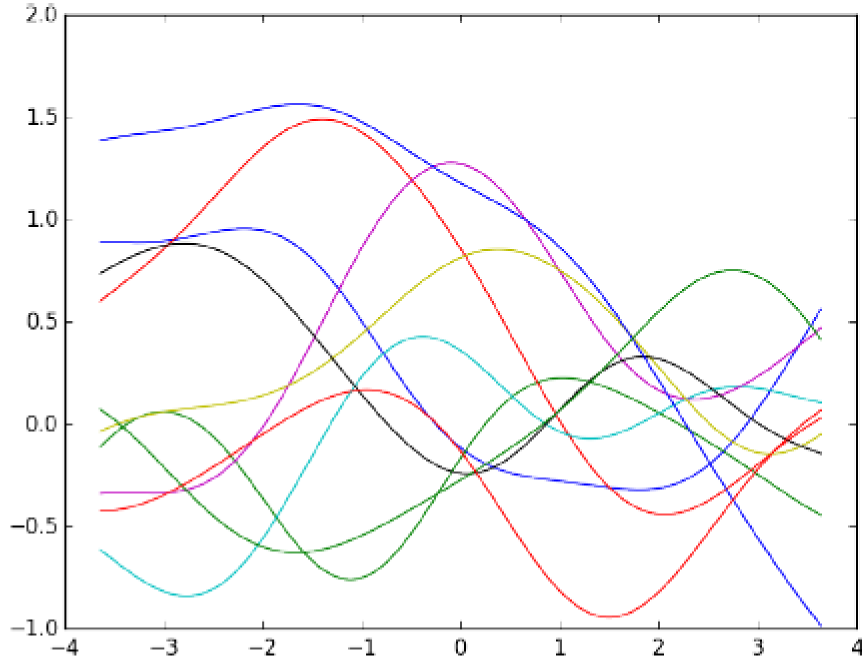
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



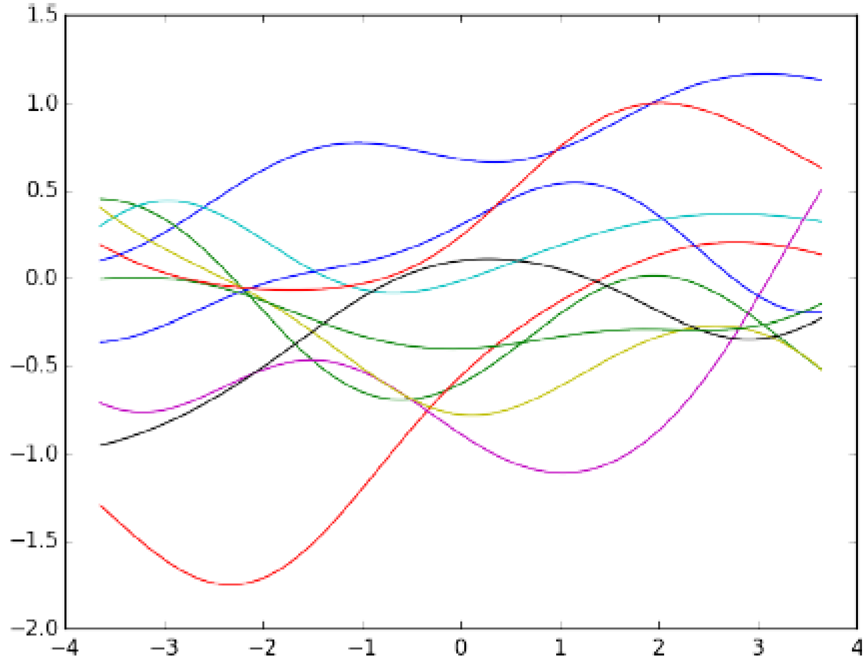
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



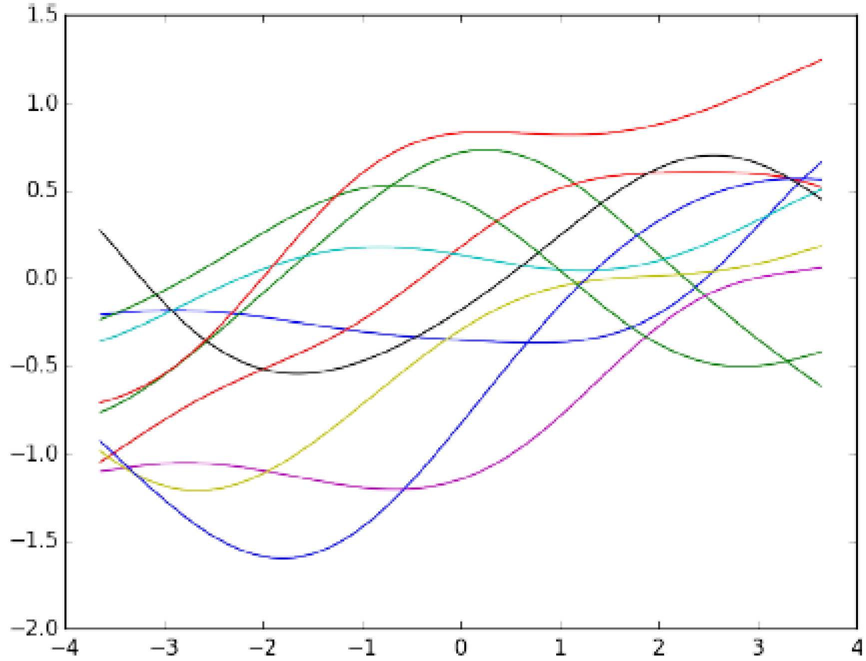
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



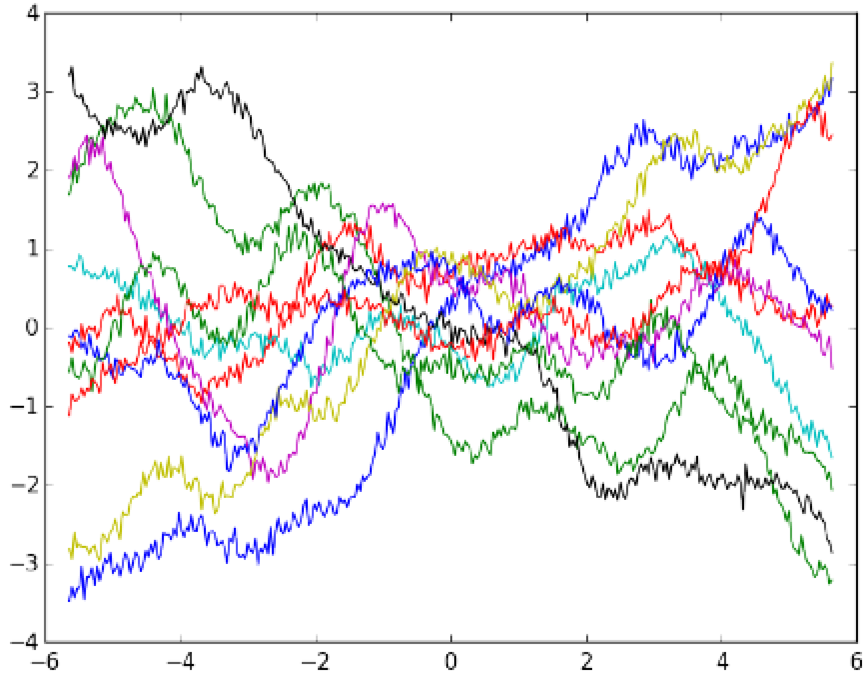
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Samples from Gaussian processes



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

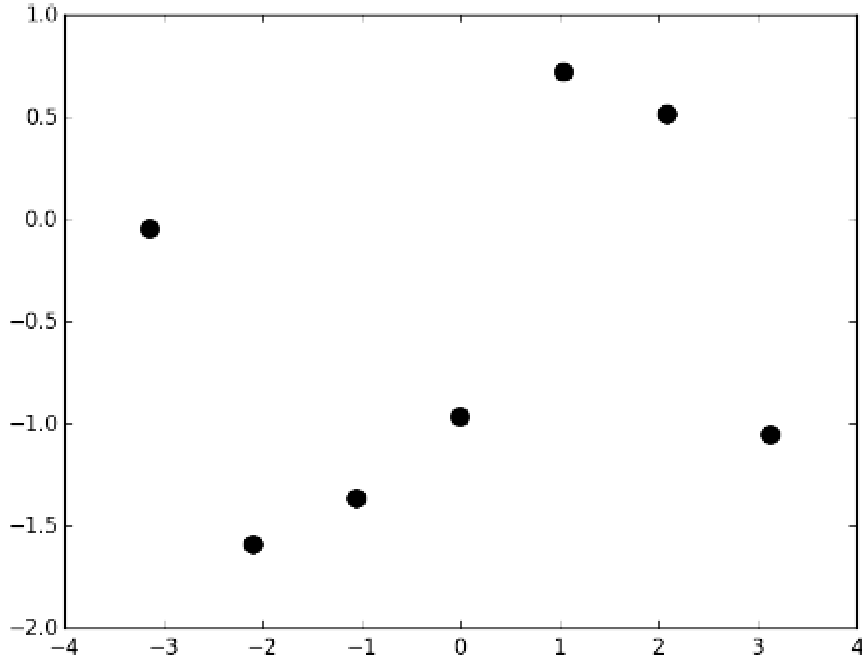
Samples from Gaussian processes



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes

This is a set of points that we know (x_i, f_i)



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, posterior

Let's predict f^* for selected $\mathbf{x}^*$ using the predictive posterior

$$p(f^* \mid \boldsymbol{x}^*, \mathbf{X}, \boldsymbol{f}, \theta)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, posterior

Let's predict f^* for selected $\mathbf{x}^*$ using the predictive posterior

First, we need to look at the joint distribution:

$$p(\boldsymbol{f}^*, \boldsymbol{f} \mid \mathbf{x}^*, \mathbf{X}, \theta) = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{x}^*) \\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, posterior

Let's predict f^* for selected $\mathbf{x}^*$ using the predictive posterior

$$p(\boldsymbol{f}^*, \boldsymbol{f} \mid \mathbf{x}^*, \mathbf{X}, \theta) = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{x}^*) \\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right)$$

$$p(\boldsymbol{f}^* \mid \mathbf{x}^*, \mathbf{X}, \boldsymbol{f}, \theta)$$

from the joint to the posterior

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, posterior

Let's predict f^* for selected $\mathbf{x}^*$ using the predictive posterior

$$p(\boldsymbol{f}^*, \boldsymbol{f} \mid \mathbf{x}^*, \mathbf{X}, \theta) = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{x}^*) \\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right)$$

from the joint to the posterior

$$p(\boldsymbol{f}^* \mid \mathbf{x}^*, \mathbf{X}, \boldsymbol{f}, \theta) =$$

$$= \mathcal{N}\left(k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} \boldsymbol{f},\ k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{x}^*) \right)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes, posterior

Let's predict f^* for selected $\mathbf{x}^*$ using the predictive posterior

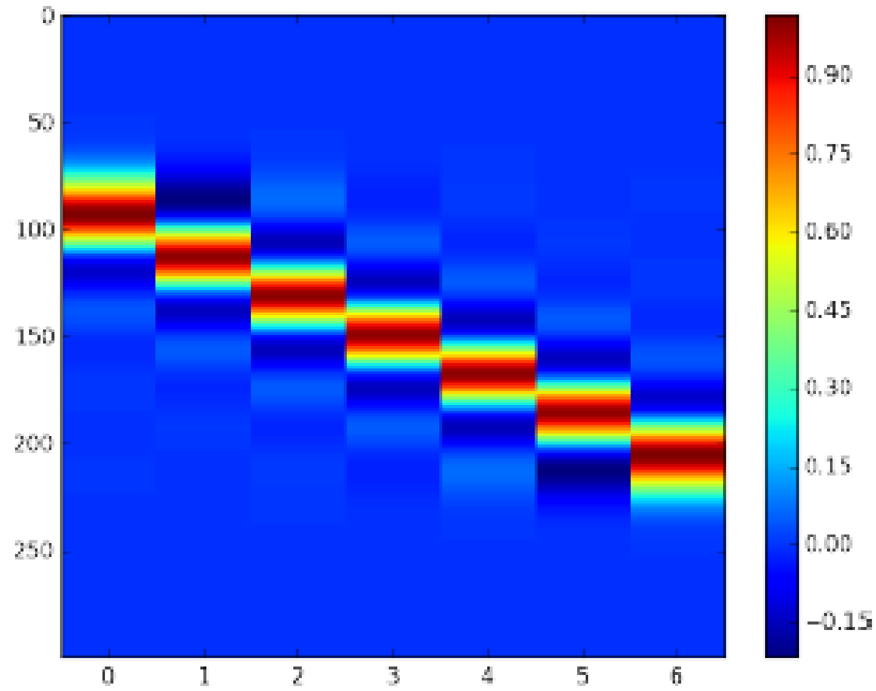
$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{x}^*) \\ k(\mathbf{x}^*,\mathbf{X}) & k(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right)$$

$$p(\boldsymbol{f}^* \mid \mathbf{x}^*, \mathbf{X}, \boldsymbol{f}, \theta) =$$

$$= \mathcal{N}\Big(\underbrace{k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} \boldsymbol{f}}_{\textit{mean}},\ \underbrace{k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{x}^*)}_{\textit{variance}} \Big)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

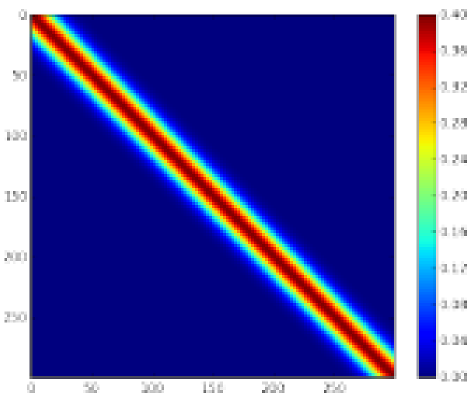
Gaussian processes



$$k(\mathbf{x}^*, \mathbf{X})^{\mathrm{T}} K(\mathbf{X}, \mathbf{X})^{-1} f$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

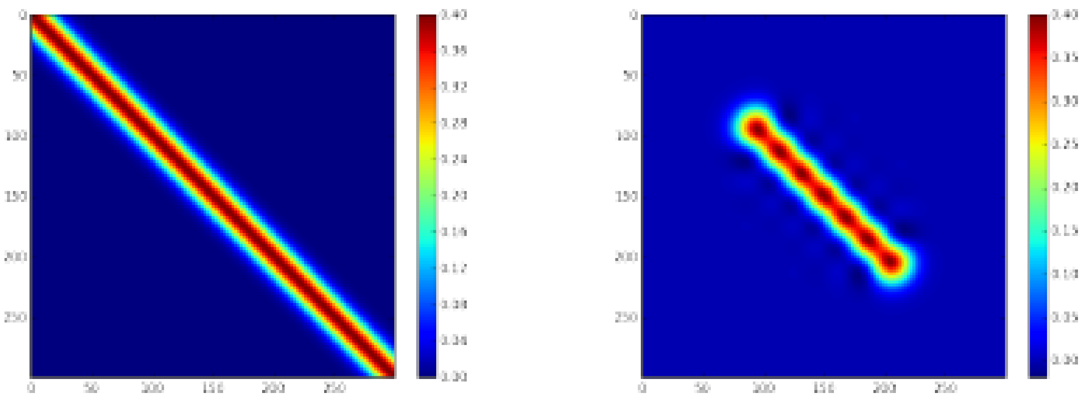
Gaussian processes



$$k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{X})^{\mathrm{T}} K(\mathbf{X}, \mathbf{X})^{-1} K(\mathbf{X}, \mathbf{x}^*)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

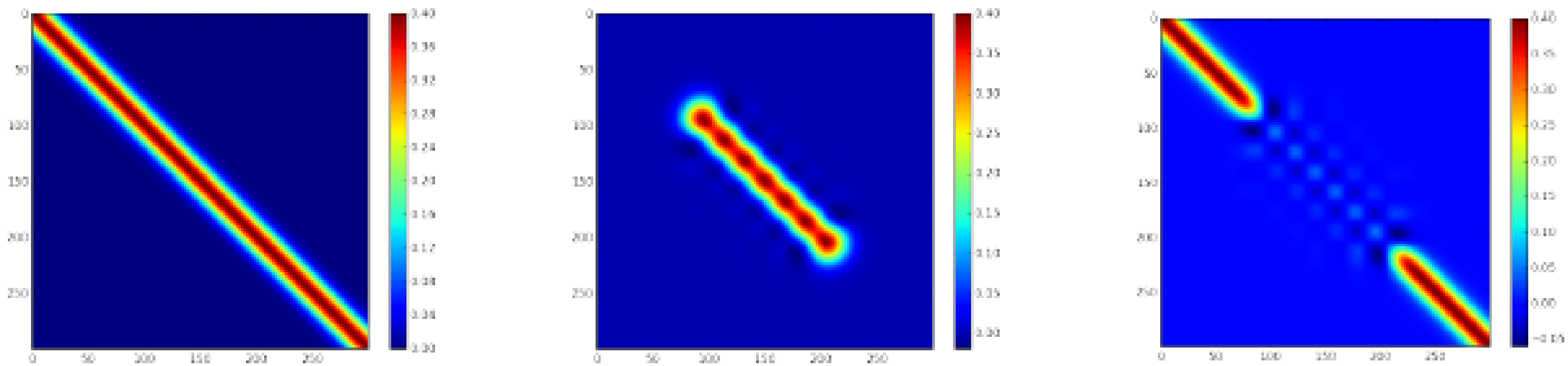
Gaussian processes



$$k(\mathbf{x}^*,\mathbf{x}^*) - \overbrace{k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{x}^*)}$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

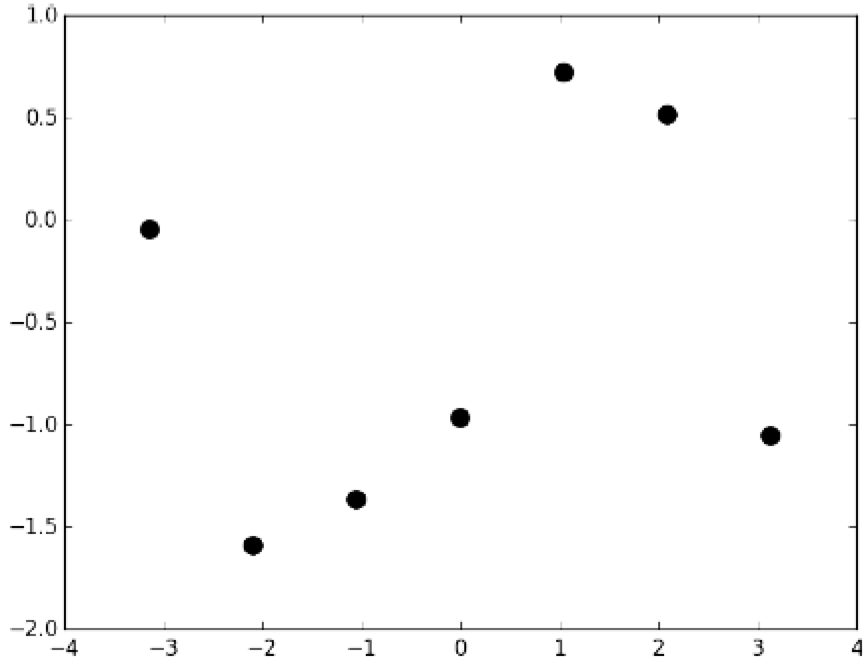
Gaussian processes



$$k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,\mathbf{X})^{\mathrm{T}} K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{x}^*)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

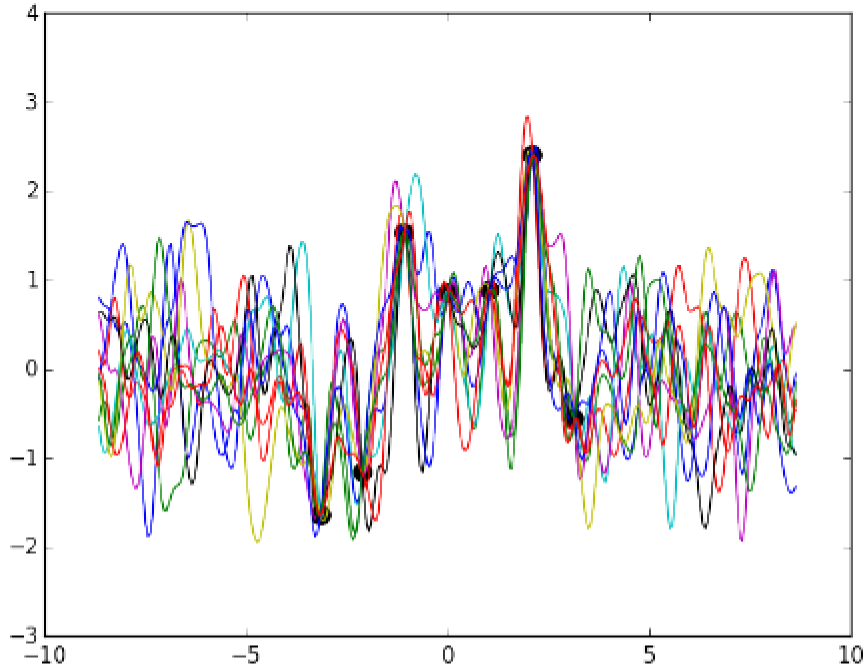
Gaussian processes



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

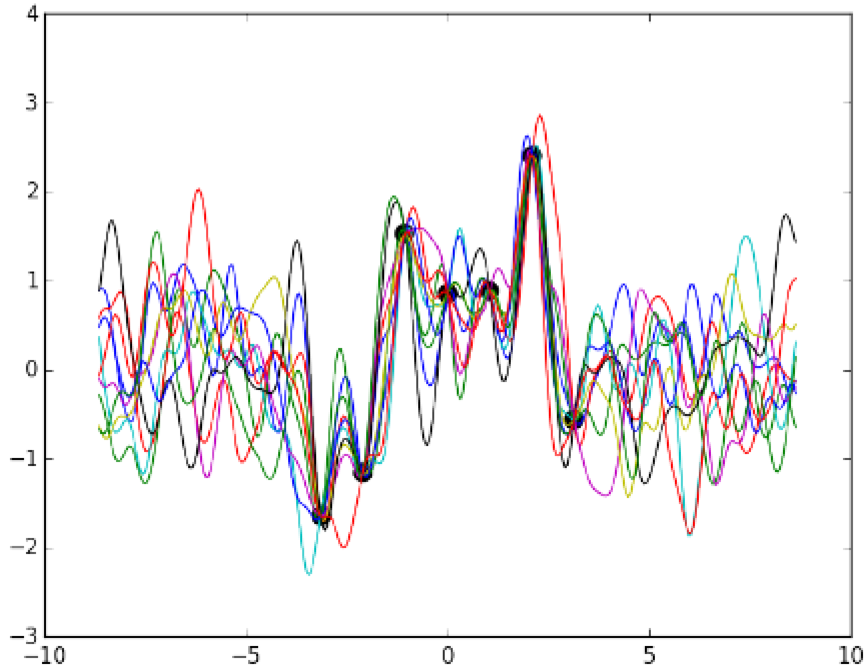
Gaussian processes

from a small length-scale parameter (little dependence between the locations)…



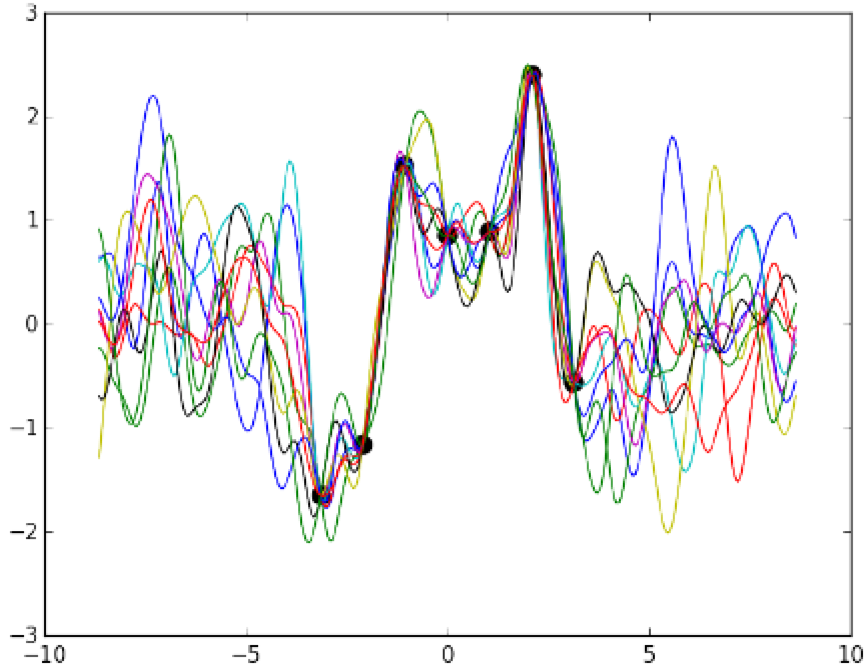
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



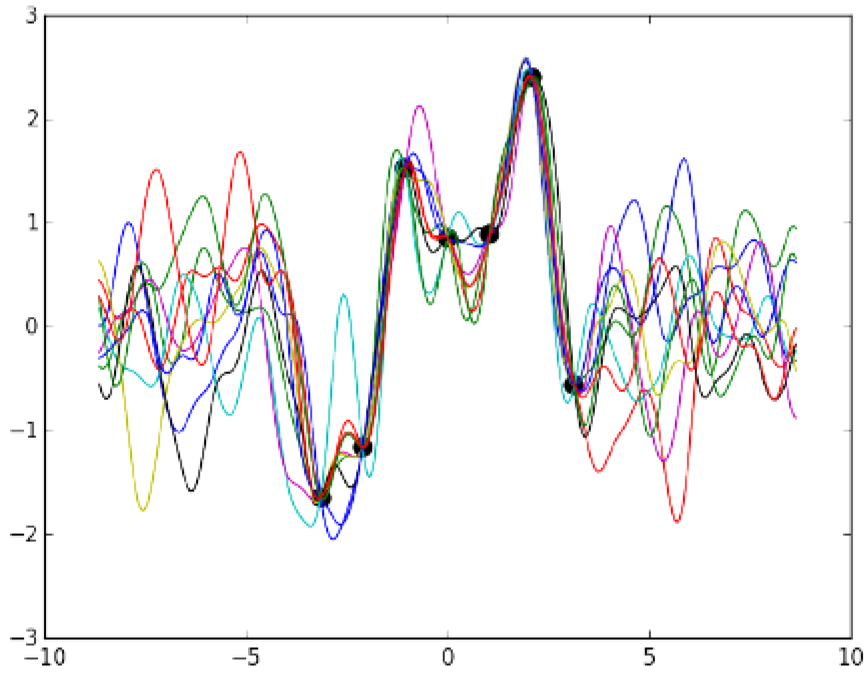
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



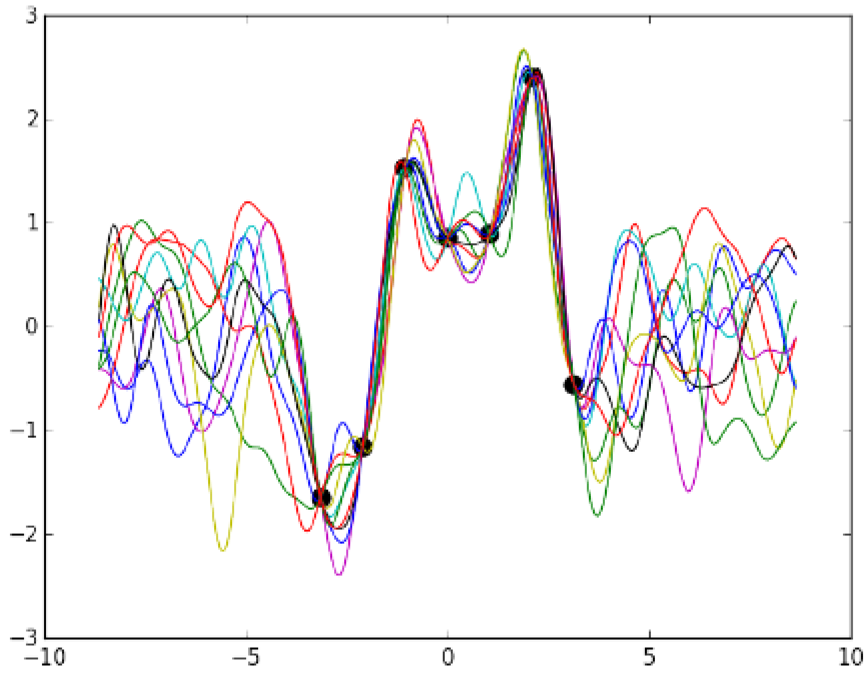
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



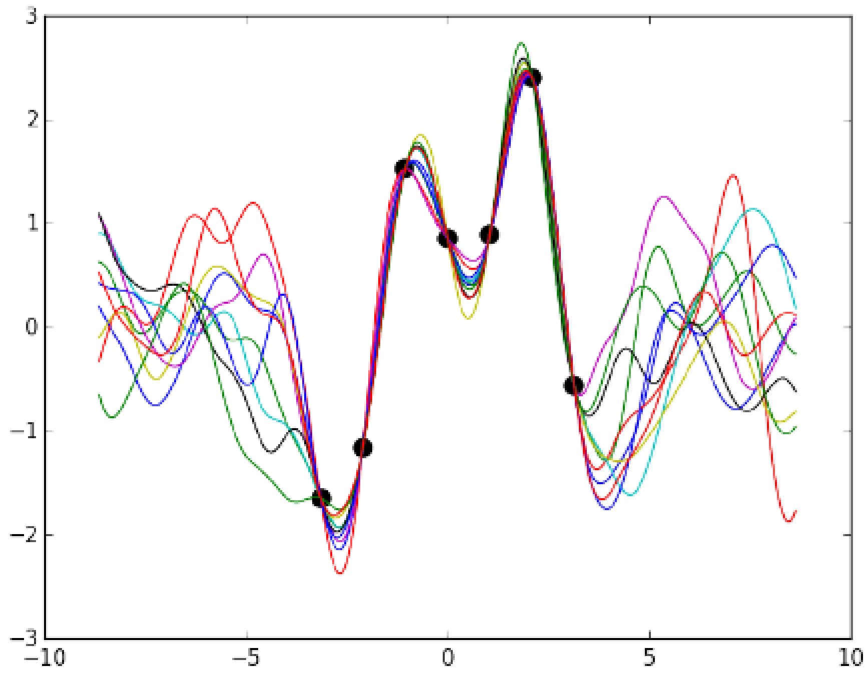
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



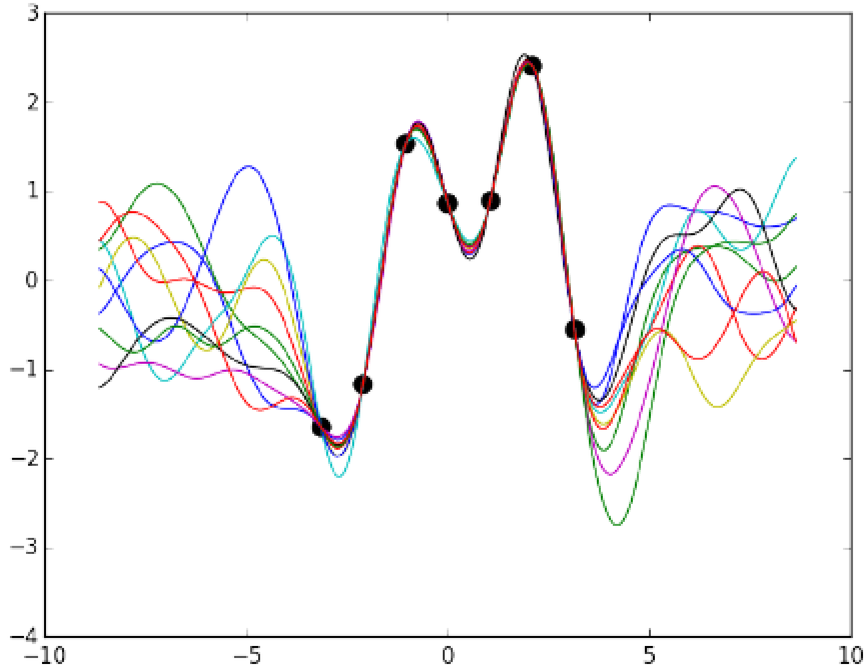
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



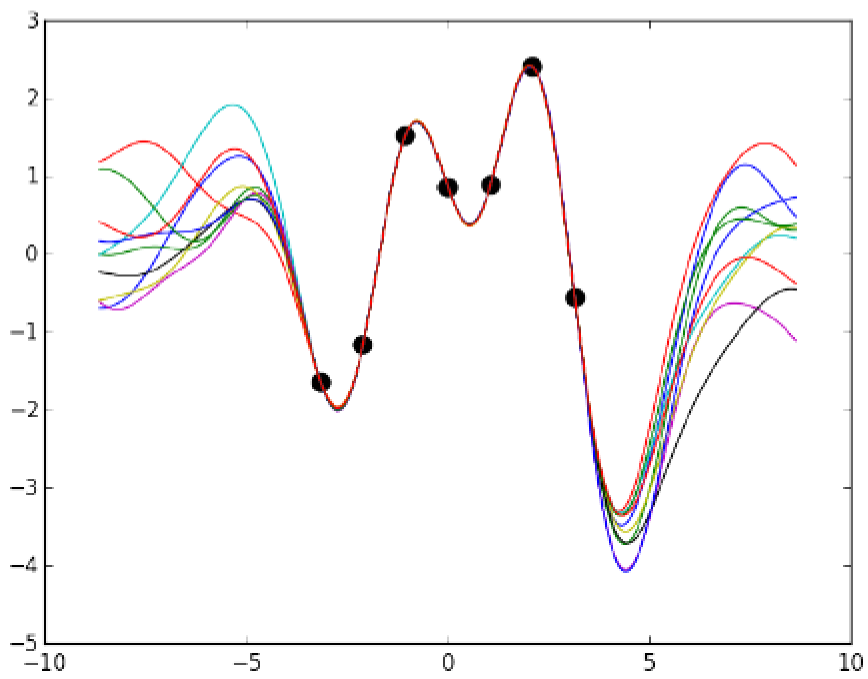
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



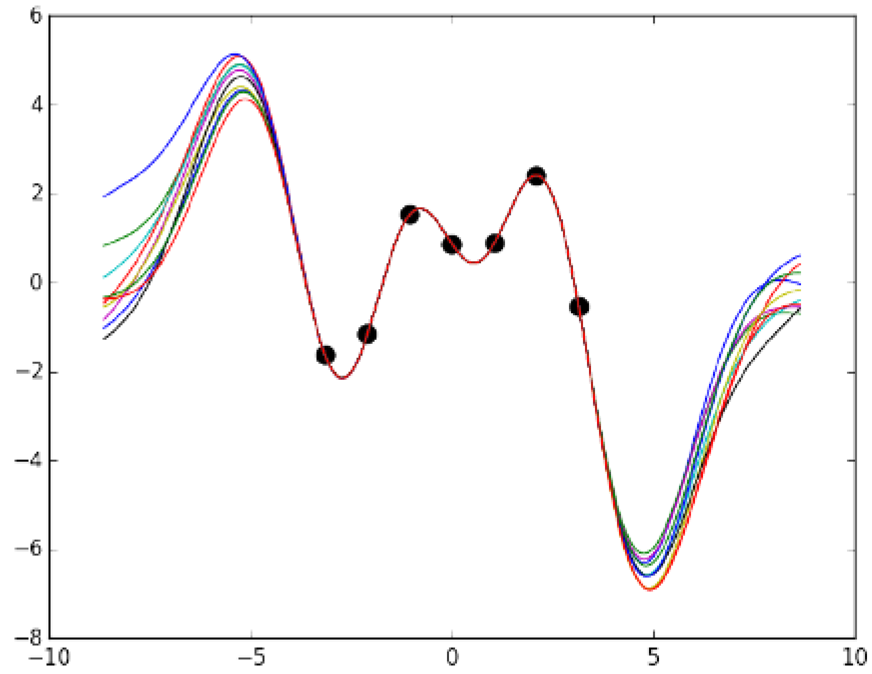
Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

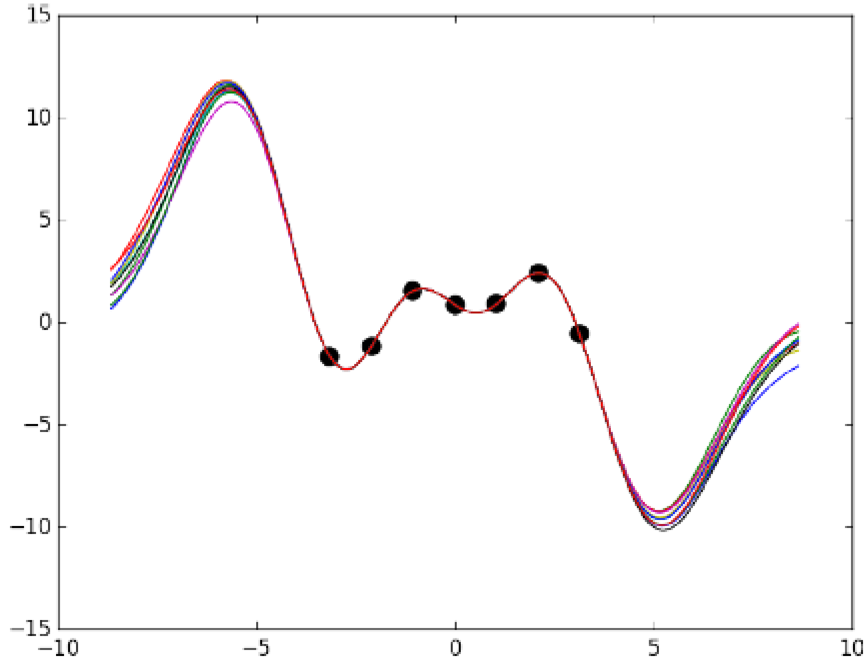
Gaussian processes



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Gaussian processes

…. to a larger length-scale parameter (more dependence between the locations)



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

In summary

- Gaussian process is a prior over function realisations
- A new random variable as the output of the mapping
- Joint distribution of any observations is Gaussian
- Posterior (predictive) distribution is conditional Gaussian

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Noise term

- So far all the realisations passed through the specified data points

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Noise term

- So far all the realisations passed through the specified data points
- Now we can assume there is noise: $t_n = f_n(x_n) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Noise term

- So far all the realisations passed through the specified data points
- Now we can assume there is noise: $t_n = f_n(x_n) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$
- The covariance matrix then becomes:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2 \delta_{ij}$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Noise term

- So far all the realisations passed through the specified data points
- Now we can assume there is noise: $t_n = f_n(x_n) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$
- The covariance matrix then becomes:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2 \delta_{ij}$$

$$\begin{bmatrix} \mathbf{t} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & k(\mathbf{X}, \mathbf{x}^*) \\ k(\mathbf{x}^*, \mathbf{X}) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right)$$

- Noise-induced independence component

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

More on covariances

An example of a periodic kernel

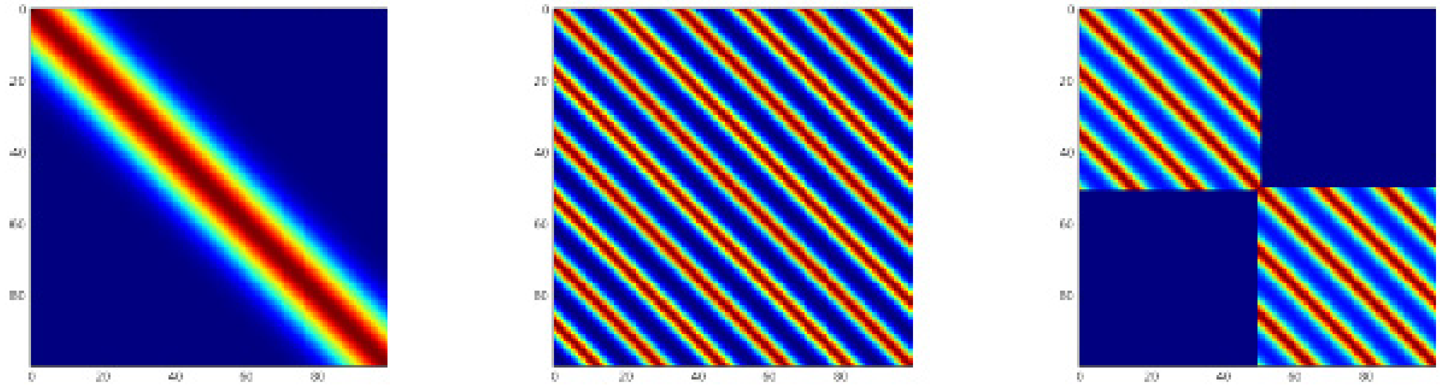
$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left\{ -\frac{2}{\ell^2} \sin^2\left(\pi \frac{|\mathbf{x}_i - \mathbf{x}_j|}{p} \right) \right\}$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

More on covariances

An example of a periodic kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \exp\left\{ -\frac{2}{\ell^2} \sin^2\left(\pi \frac{|\boldsymbol{x}_i - \boldsymbol{x}_j|}{p} \right) \right\}$$



Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

More on covariances

To summarise

- Covariance function encodes a preference for function behaviour
- Selecting the right covariance is a very important task
- Encode whatever you know about (co-)variations in data

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Hyperparameters

- Prior has parameters (describe the covariance matrix $\mathbf{K}$ and noise)
- How do we set them up?
  - we could make assumptions and fix them

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Hyperparameters

- Prior has parameters (describe the covariance matrix $\mathbf{K}$ and noise)
- How do we set them up?
  - we could make assumptions and fix them
  - but it is more suitable to see what the data has to say -> learning from data

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Hyperparameters

- Prior has parameters (describe the covariance matrix $\mathbf{K}$ and noise)
- How do we set them up?
  - we could make assumptions and fix them
  - but it is more suitable to see what the data has to say -> learning from data
  - learning in Gaussian processes = inferring hyperparameters from model using data

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Marginal likelihood

$$p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta}) = \int \underbrace{p(\mathbf{t} \mid \boldsymbol{f})}_{\text{likelihood}} \underbrace{p(\boldsymbol{f} \mid \mathbf{X}, \boldsymbol{\theta})}_{\text{prior}} d\boldsymbol{f}$$

- we are not interested directly in $\boldsymbol{f}$ so we should *marginalise* it out
- importantly, Gaussian marginal is Gaussian (the integral)

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Marginal likelihood

$$p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta}) = \int \underbrace{p(\mathbf{t} \mid \boldsymbol{f})}_{\text{likelihood}} \underbrace{p(\boldsymbol{f} \mid \mathbf{X}, \boldsymbol{\theta})}_{\text{prior}} d\boldsymbol{f}$$

- how to deal with $\boldsymbol{\theta}$?........ Bayesian approach would be best, i.e. identify the prior, posterior and average it out

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

Marginal likelihood

$$p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta}) = \int \underbrace{p(\mathbf{t} \mid \boldsymbol{f})}_{\text{likelihood}} \underbrace{p(\boldsymbol{f} \mid \mathbf{X}, \boldsymbol{\theta})}_{\text{prior}} d\boldsymbol{f}$$

- how to deal with $\boldsymbol{\theta}$?........ Bayesian approach would be best, i.e.

  identify the prior, posterior and average it out
- BUT: computational heavy -> need for heavy variational approach

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

- We resort to Type II Maximum Likelihood

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta})$$

  - different from a classical ML estimate (marginalisation of f first)
  - let's minimise the negative logarithm

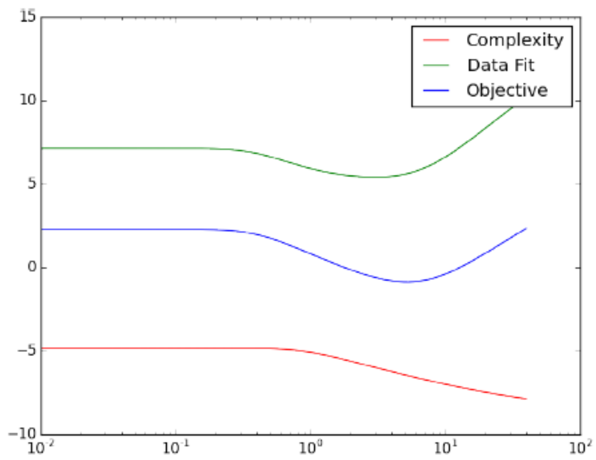
$$\arg\max_{\boldsymbol{\theta}} p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \left\{ -\ln p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\theta}) \right\}$$

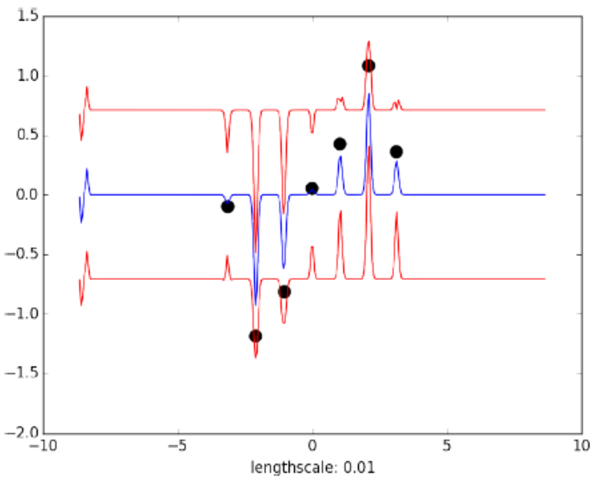
$$\Downarrow$$

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{2} \mathbf{t}^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{t} + \frac{1}{2} \ln |\mathbf{K}| + \frac{N}{2} \ln(2\pi) \right\}$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes



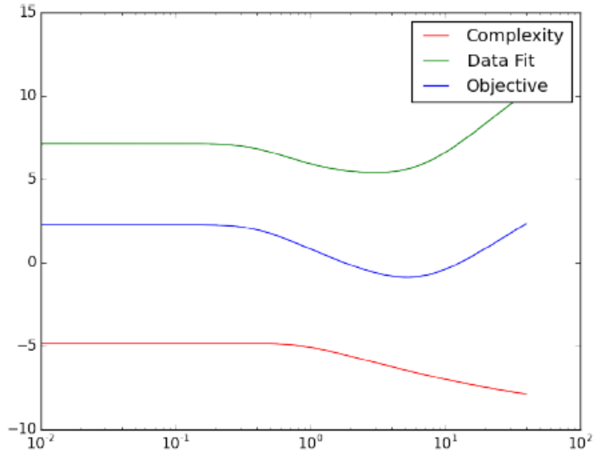


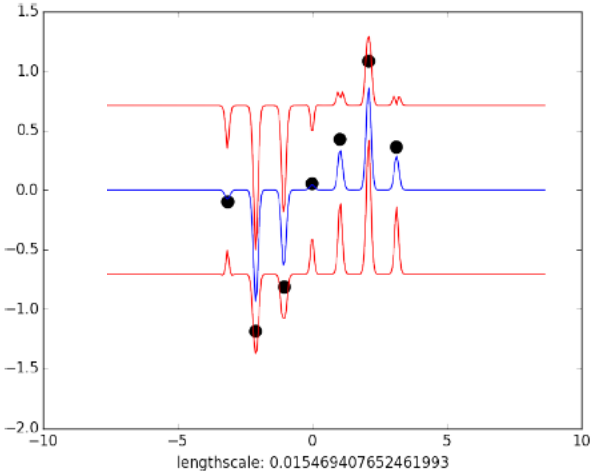
Data Fit
Complexity
Objective

$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

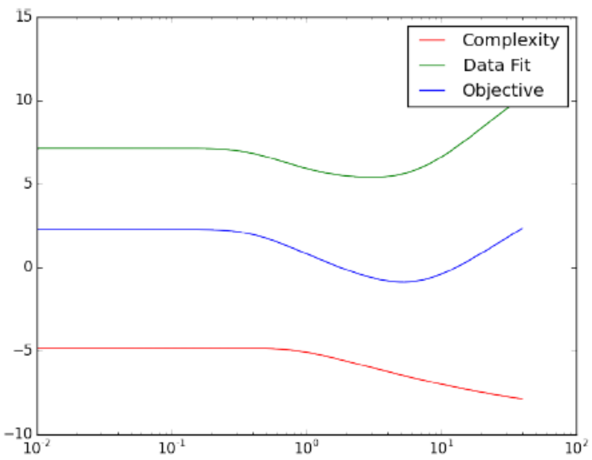


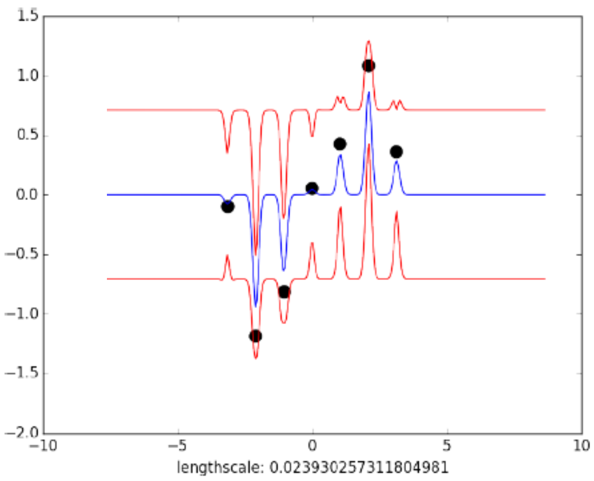


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

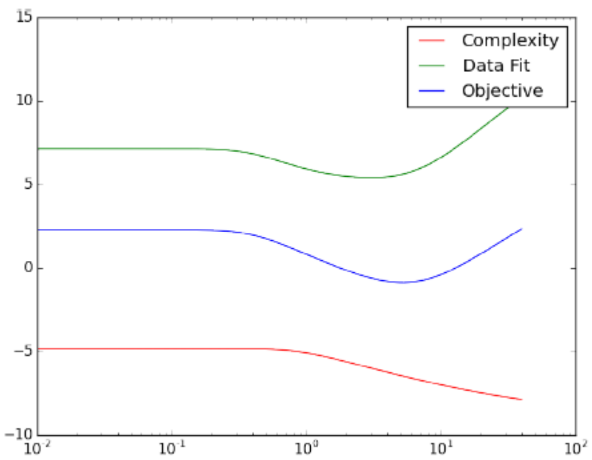


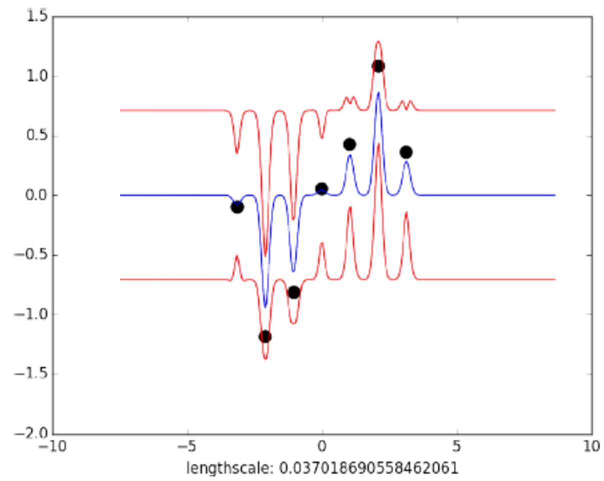


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

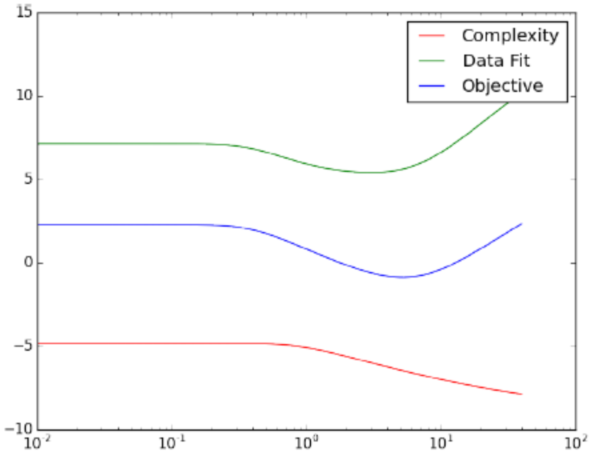


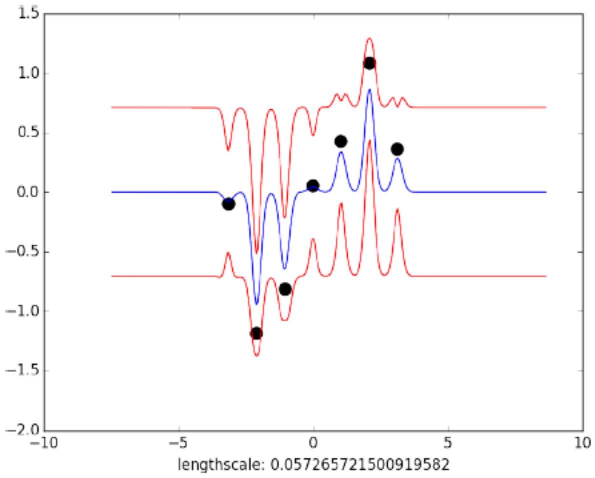


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

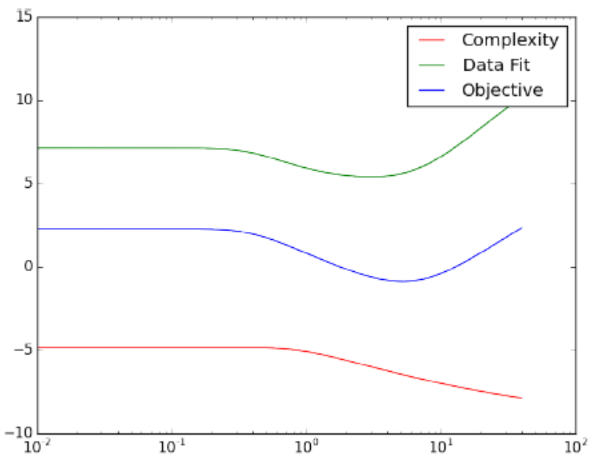


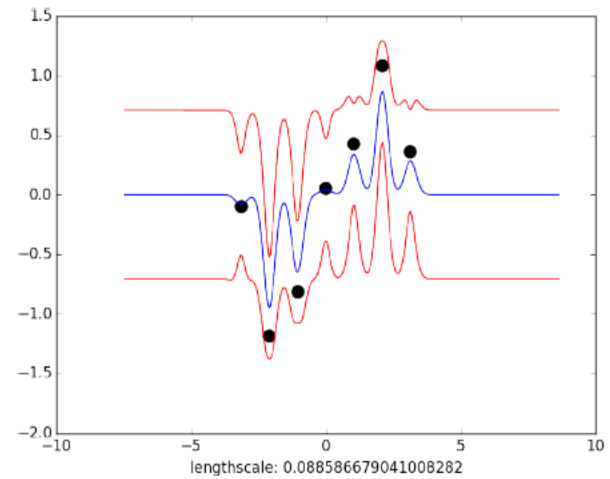


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

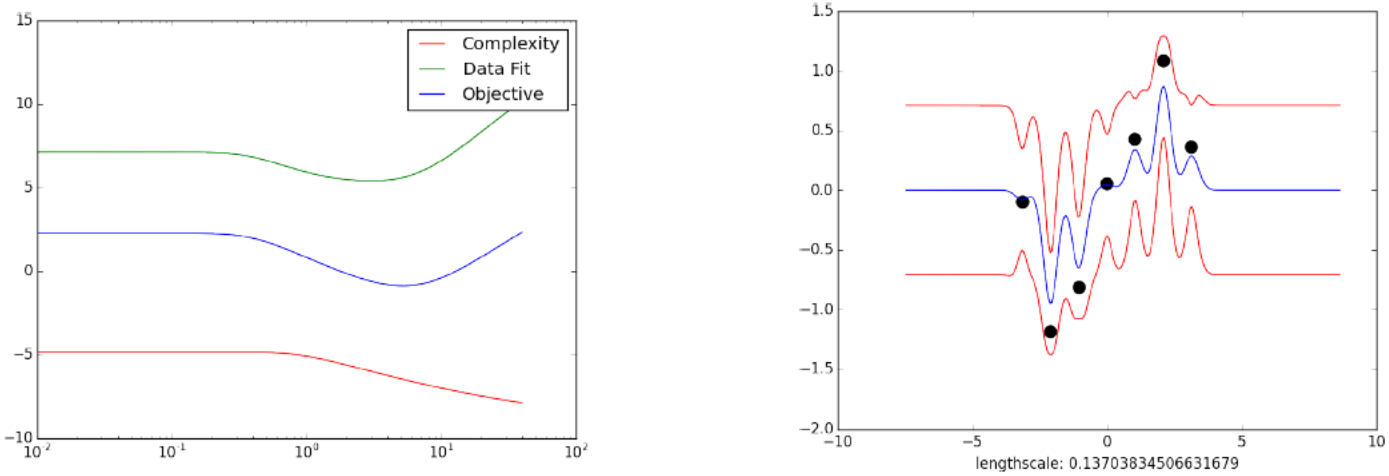




$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

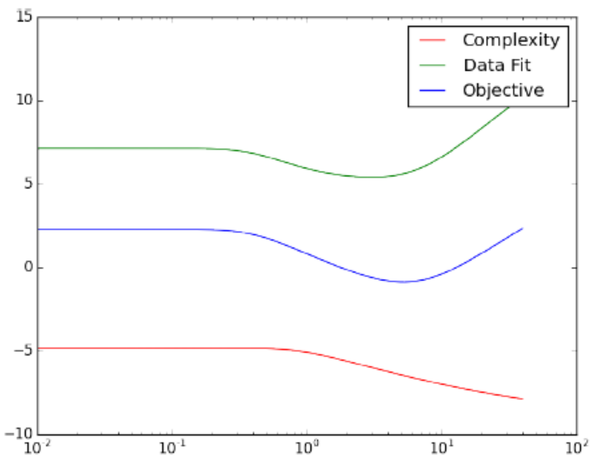
Learning in Gaussian processes

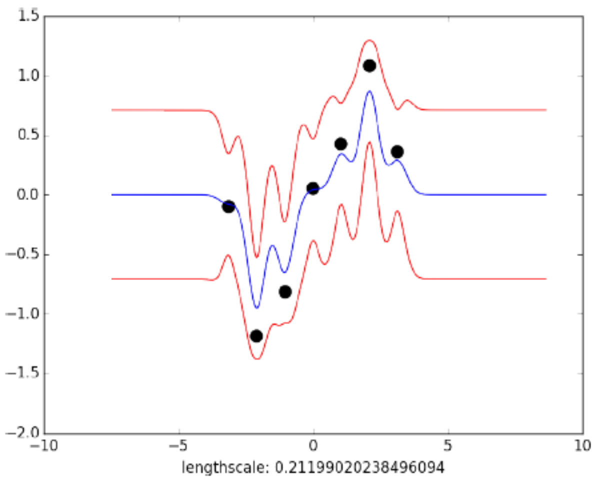


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

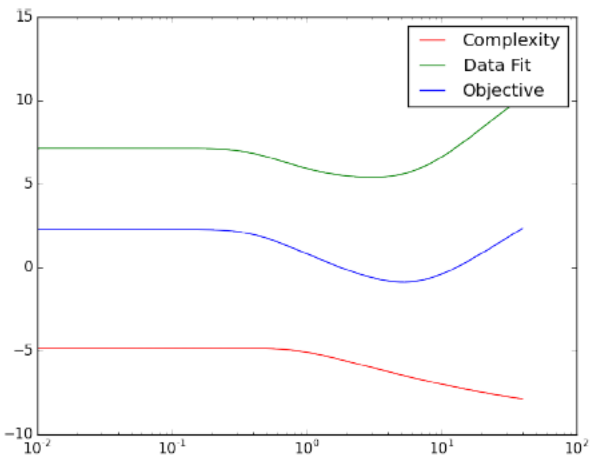


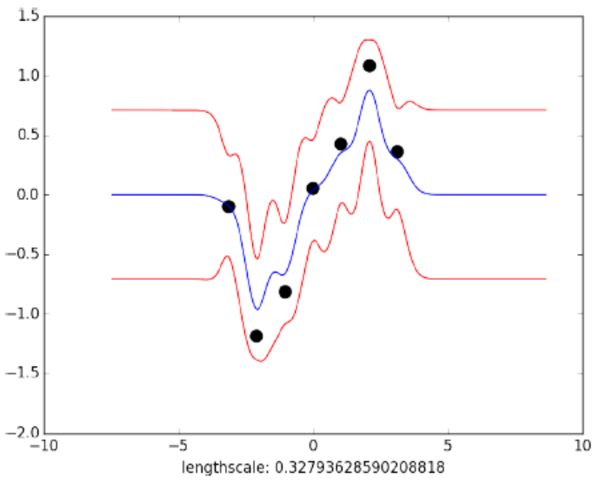


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

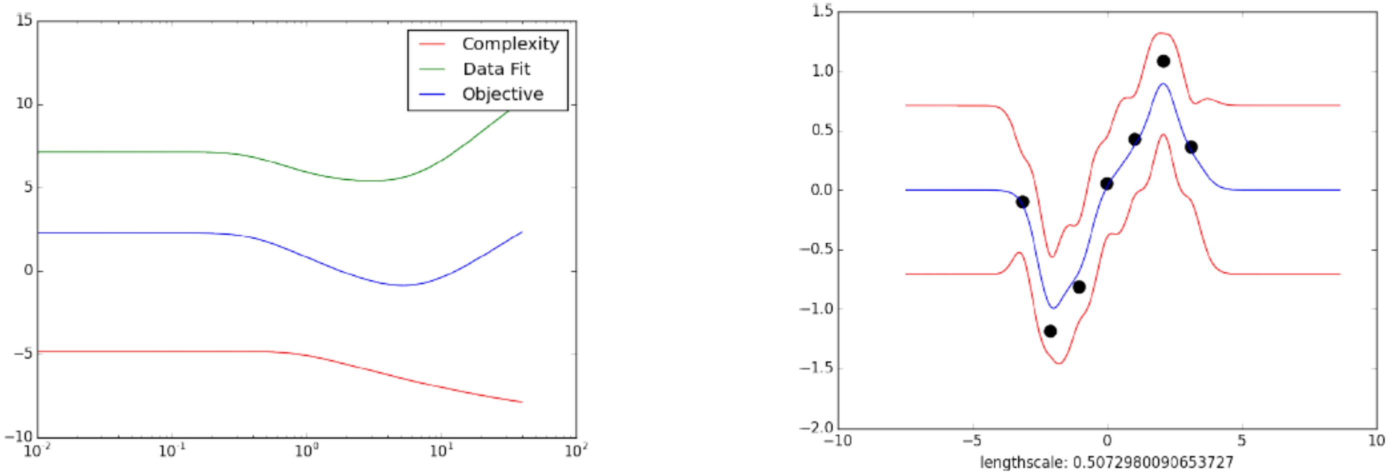




$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

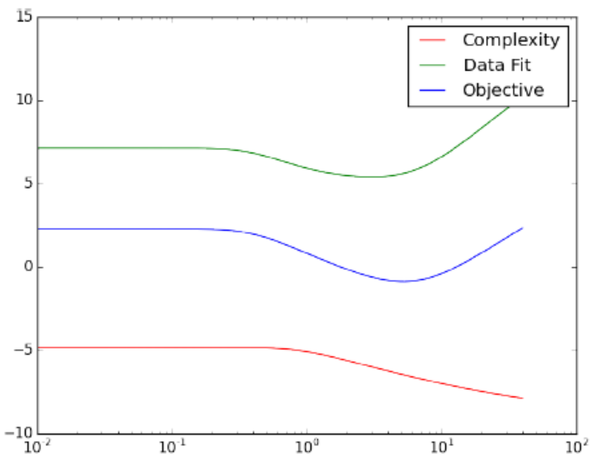
Learning in Gaussian processes

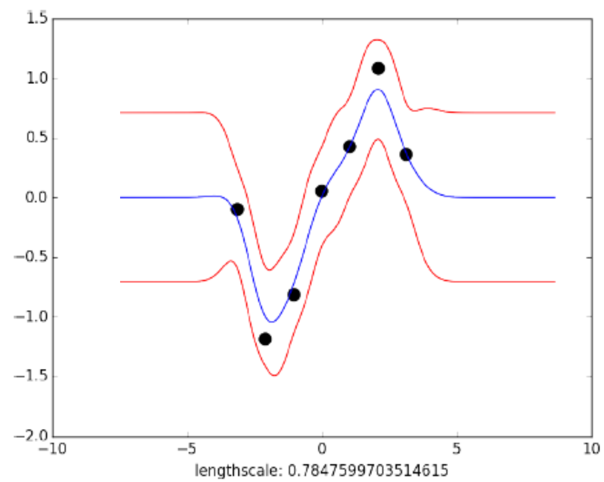


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

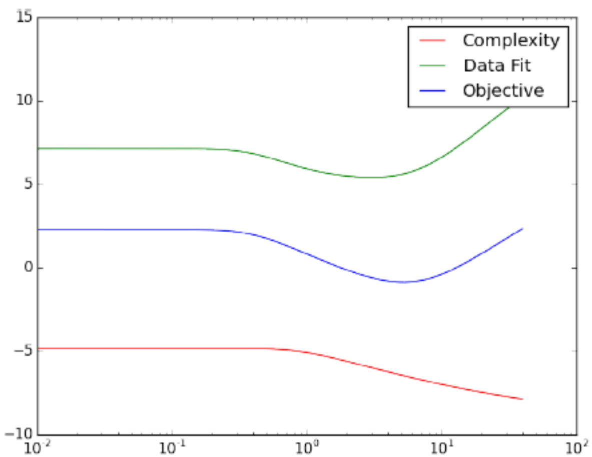


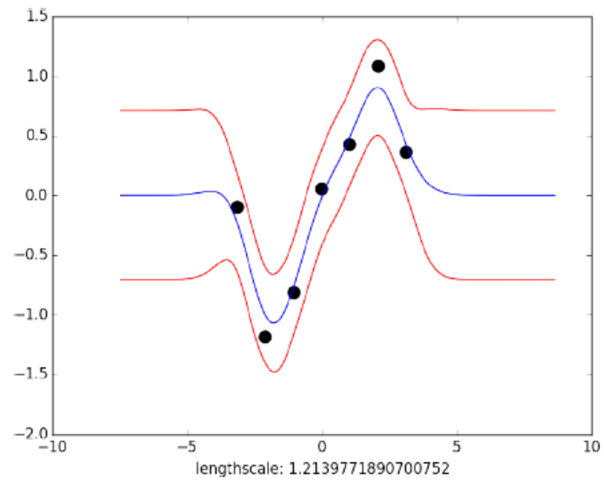


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

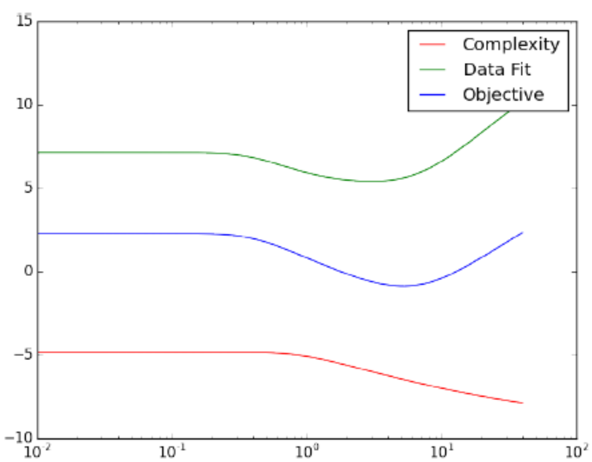


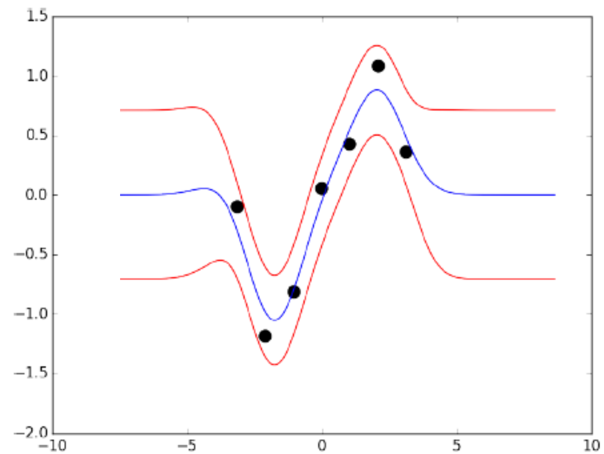


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

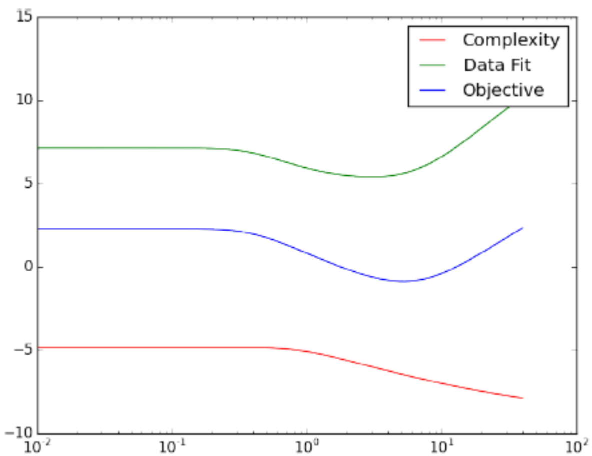


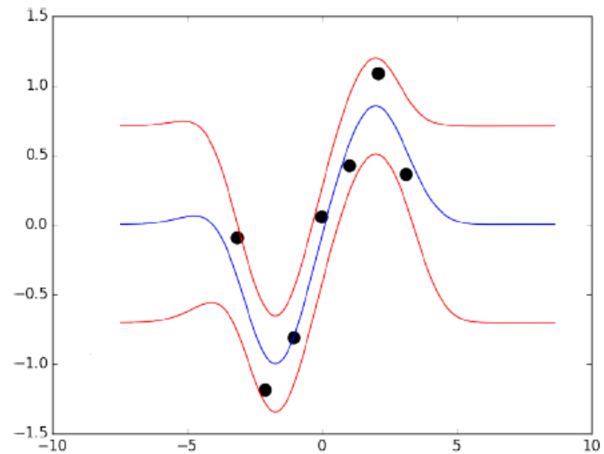


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

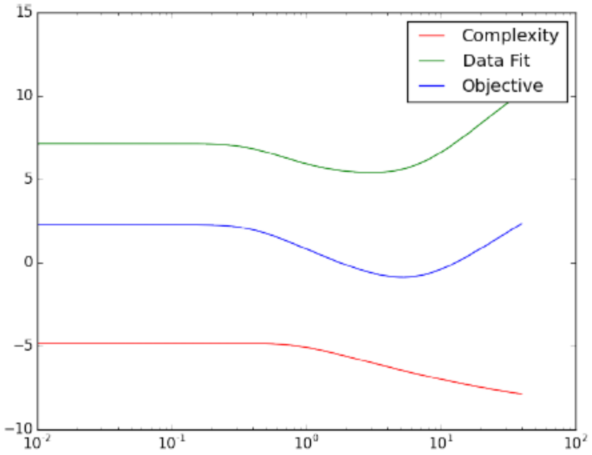


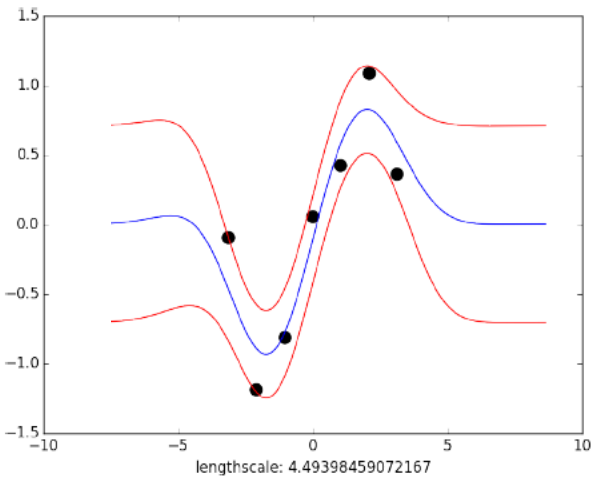


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

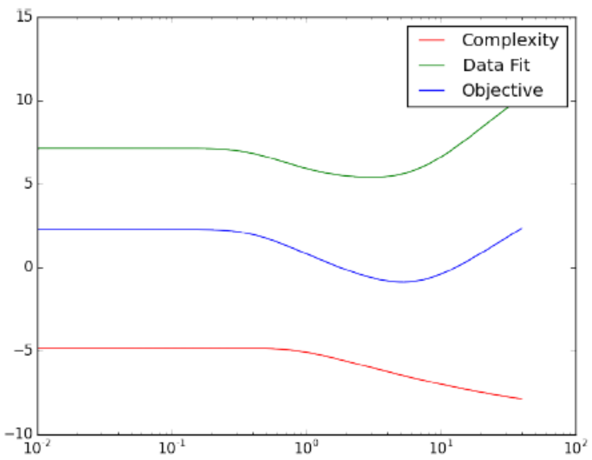


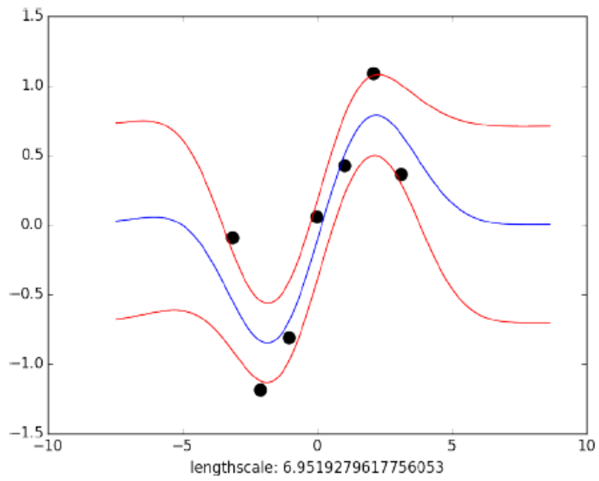


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

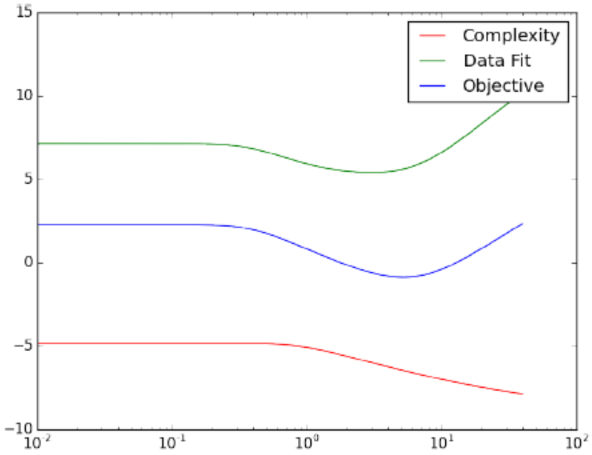


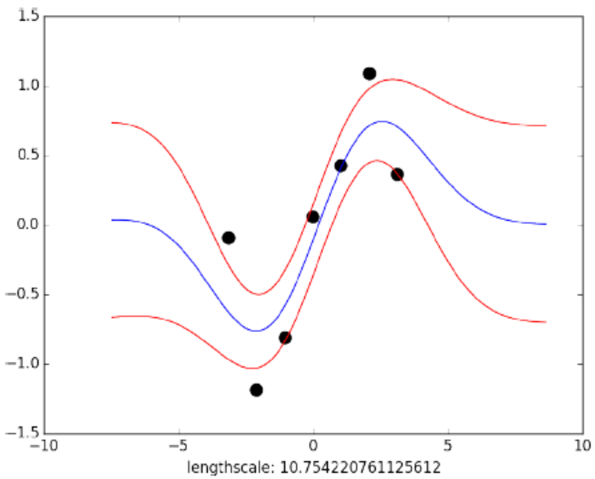


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

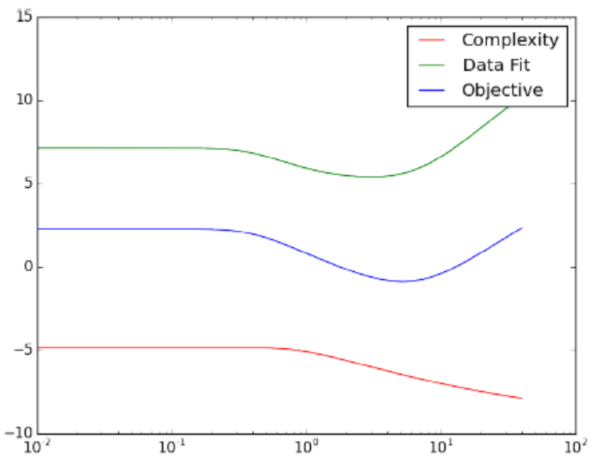


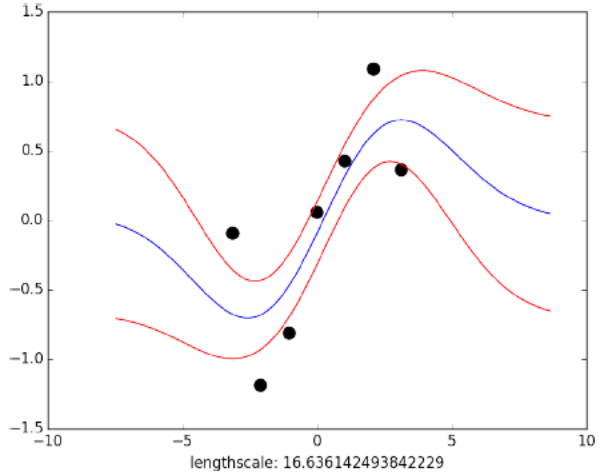


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

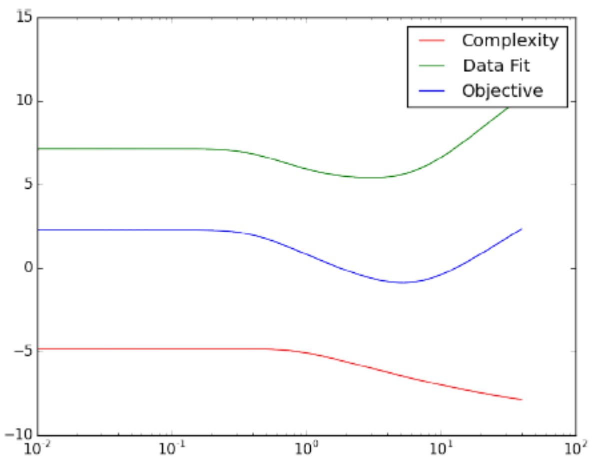


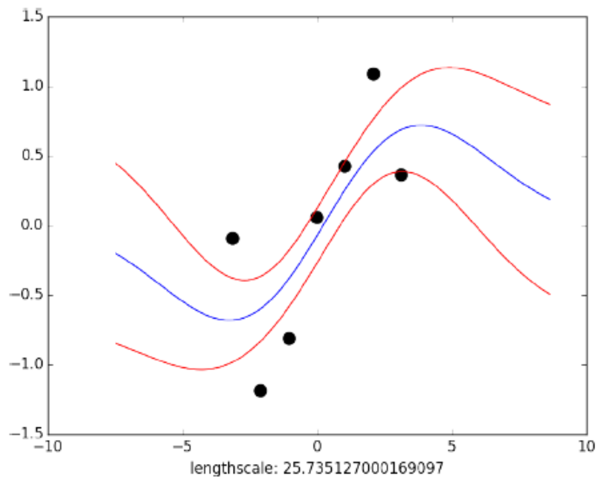


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

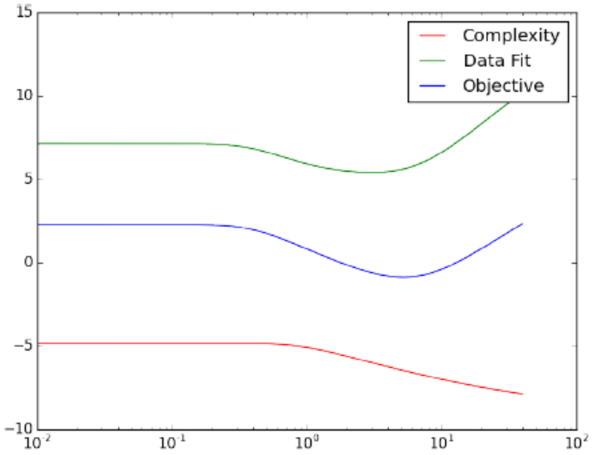


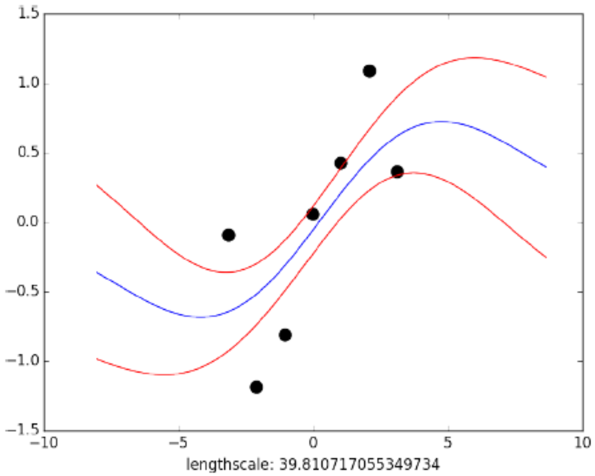


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes

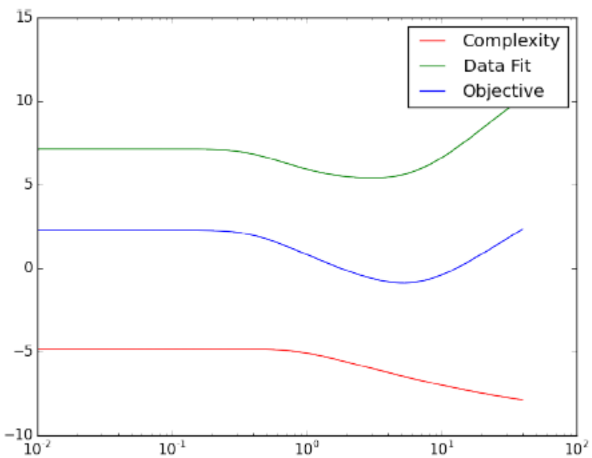


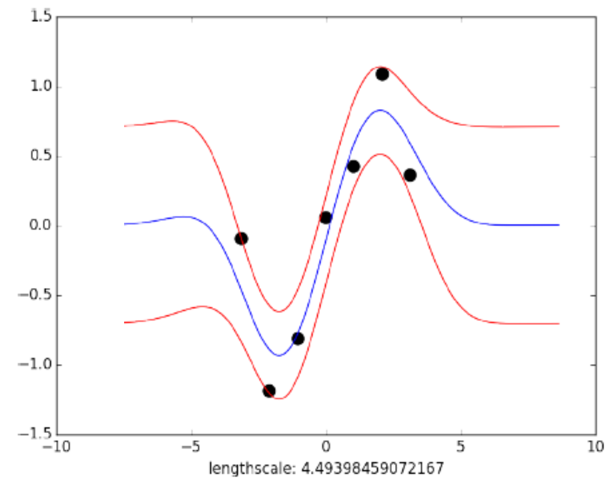


$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} + \frac{1}{2}\log\left|\mathbf{K}\right| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Learning in Gaussian processes





$$L(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{y} + \frac{1}{2}\log|\mathbf{K}| + \frac{1}{2}\log(2\pi)$$

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Summary

- Kernels as covariance between data points
- The feature space implied by kernel does not have to be explicitly known and can have infinite dimensionality
- Kernel trick allows for nonlinear mapping with linear methods
- Gaussian processes are priors over functions
- Predictive (posterior) distribution is conditional Gaussian
- Gaussian processes provide scope for averaging over all possible functions
- Bayesian inference as before, just a different prior

Adapted from Carl Henrik Ek's lecture (DD2434, 2015)

Recommended additional reading

On kernels

Schölkopf and Smola (2002) *Learning with Kernels*.

Shawe-Taylor and Cristianini (2004) *Kernel Methods for Pattern Analysis*.

On Gaussian processes

Rasmussen and Williams(2006) *Gaussian Processes for Machine Learning*.